

BUT SD 3

DOSSIER NOSQL

2024

Lafaille Virgile
Nasseh Freitas Enrique
Kayadjianian Théo

Sommaire

Table des matières

<i>Sommaire</i>	2
<i>Introduction</i>	3
<i>Données</i>	3
<i>SQL</i>	3
<i>Import des fichiers</i>	7
<i>Importation des tables</i>	12
<i>Création de relations</i>	19
<i>Requête de visualisation</i>	24
<i>Requête en réponse à la problématique</i>	26
<i>Conclusion:</i>	27

Introduction

Dans le cadre de notre projet d'étude la tâche qui nous a été confié consiste en l'intégration de données d'après un environnement Nosql. L'angle d'attaque qui nous a été proposé par le prestataire est l'étude d'un événement sportif.

Nous avons choisi de parler de l'euro de football 2016. Nous avons décliné la problématique suivante qui servira de fil conducteur pour l'ensemble de nos décisions, de nos analyses et de nos résultats. Qu'elle a été l'impact de l'euro 2016 de football masculin sur les infrastructures sportives et le nombre de licenciés ?

Bien que ce projet soit focalisé sur un événement sportif l'objectif de ce travail est de produire un « modèle » pouvant être réutilisé sur des données plus anciennes ou mis à jour avec des données plus récentes.

Données

Pour pouvoir mener ce projet à bien nous avons dû sélectionner des données en ligne. Dans où soucis de simplicité et de praticité nous avons nous même créer certaines tables en croisant différents jeux de données.

Infrastructures : données gouvernementales

Licencié: données condensé à la main

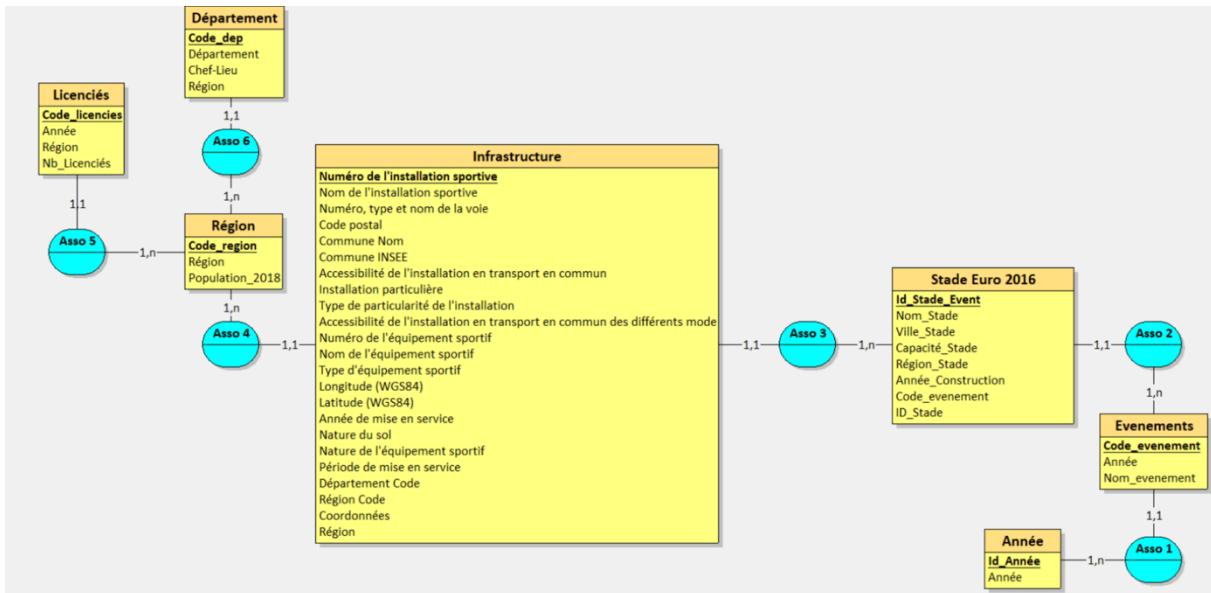
SQL

Nous avons décidé d'étudier le problème d'une manière relationnelle en important nos données dans un environnement SQL

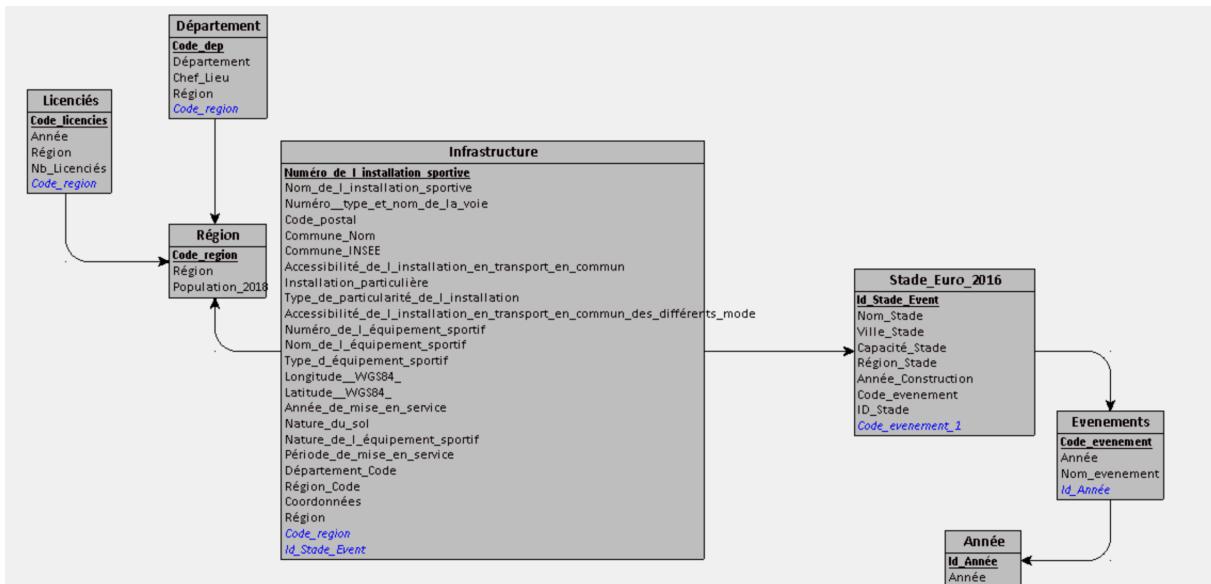
Dans notre cas nous avons fait le choix de partir sur DB browser par sa simplicité d'utilisation. Nous sommes parties du principe que passer sur un environnement SQL nous permettrait de mieux apprendre les relations entre les tables afin de pourvoir importer nos différentes données de manière optimale dans neo4j. Pour ce faire nous avons réalisé notre modèle conceptuel de données sur le logiciel (MCD) et notre modèle logique de données (MLD) sur ce logiciel nous avons pu mettre en place nos relations entre table, l'avantage de ce logiciel réside dans sa capacité à générer un code SQL qui facilite grandement l'intégration de nos données dans DB browser.

MCD

Ce MCD nous permet d'avoir une première visualisation de nos relations, et une vue globale de nos tables et de nos colonnes. Pour des raisons de simplicité nous avons intégré nos données dans DB browser pour se faire une première idée des résultats à fournir



MLD :



Code pour les BDD SQL :

```
CREATE TABLE Région(
    Code_region INT,
    Région VARCHAR(50),
    Population_2018 INT,
    PRIMARY KEY(Code_region)
);

CREATE TABLE Licenciés(
    Code_licences INT,
    Année INT,
    Région VARCHAR(50),
    Nb_Licenciés INT,
    Code_region INT NOT NULL,
    PRIMARY KEY(Code_licences),
    FOREIGN KEY(Code_region) REFERENCES Région(Code_region)
);

CREATE TABLE Année(
    Id_Année INT,
    Année INT,
    PRIMARY KEY(Id_Année)
);

CREATE TABLE Département(
    Code_dep INT,
    Département VARCHAR(50),
    Chef_Lieu VARCHAR(50),
    Région VARCHAR(50),
    Code_region INT NOT NULL,
    PRIMARY KEY(Code_dep),
    FOREIGN KEY(Code_region) REFERENCES Région(Code_region)
);

CREATE TABLE Evenements(
    Code_evenement INT,
    Année INT,
    Nom_evenement VARCHAR(50),
    Id_Année INT NOT NULL,
    PRIMARY KEY(Code_evenement),
    FOREIGN KEY(Id_Année) REFERENCES Année(Id_Année)
);

CREATE TABLE Stade_Euro_2016(
    Id_Stade_Event INT,
```

```

Nom_Stade VARCHAR(50),
Ville_Stade VARCHAR(50),
Capacité_Stade INT,
Région_Stade VARCHAR(50),
Année_Construction INT,
Code_evenement INT,
ID_Stade INT,
Code_evenement_1 INT NOT NULL,
PRIMARY KEY(Id_Stade_Event),
FOREIGN KEY(Code_evenement_1) REFERENCES Evenements(Code_evenement)
);

```

```

CREATE TABLE Infrastructure(
Numéro_de_l_installation_sportive VARCHAR(50),
Nom_de_l_installation_sportive VARCHAR(50),
Numéro_type_et_nom_de_la_voie VARCHAR(50),
Code_postal INT,
Commune_Nom VARCHAR(50),
Commune_INSEE INT,
Accessibilité_de_l_installation_en_transport_en_commun LOGICAL,
Installation_particulière LOGICAL,
Type_de_particularité_de_l_installation VARCHAR(50),
Accessibilité_de_l_installation_en_transport_en_commun_des_différents_mode
VARCHAR(50),
Numéro_de_l équipement_sportif VARCHAR(50),
Nom_de_l équipement_sportif VARCHAR(50),
Type_d équipement_sportif VARCHAR(50),
Longitude_WGS84_DECIMAL(15,10),
Latitude_WGS84_DECIMAL(15,10),
Année_de_mise_en_service INT,
Nature_du_sol VARCHAR(50),
Nature_de_l équipement_sportif VARCHAR(50),
Période_de_mise_en_service VARCHAR(50),
Département_Code INT,
Région_Code INT,
Coordonnées VARCHAR(50),
Région VARCHAR(50),
Code_region INT NOT NULL,
Id_Stade_Event INT NOT NULL,
PRIMARY KEY(Numéro_de_l_installation_sportive),
FOREIGN KEY(Code_region) REFERENCES Région(Code_region),
FOREIGN KEY(Id_Stade_Event) REFERENCES Stade_Euro_2016(Id_Stade_Event)
);

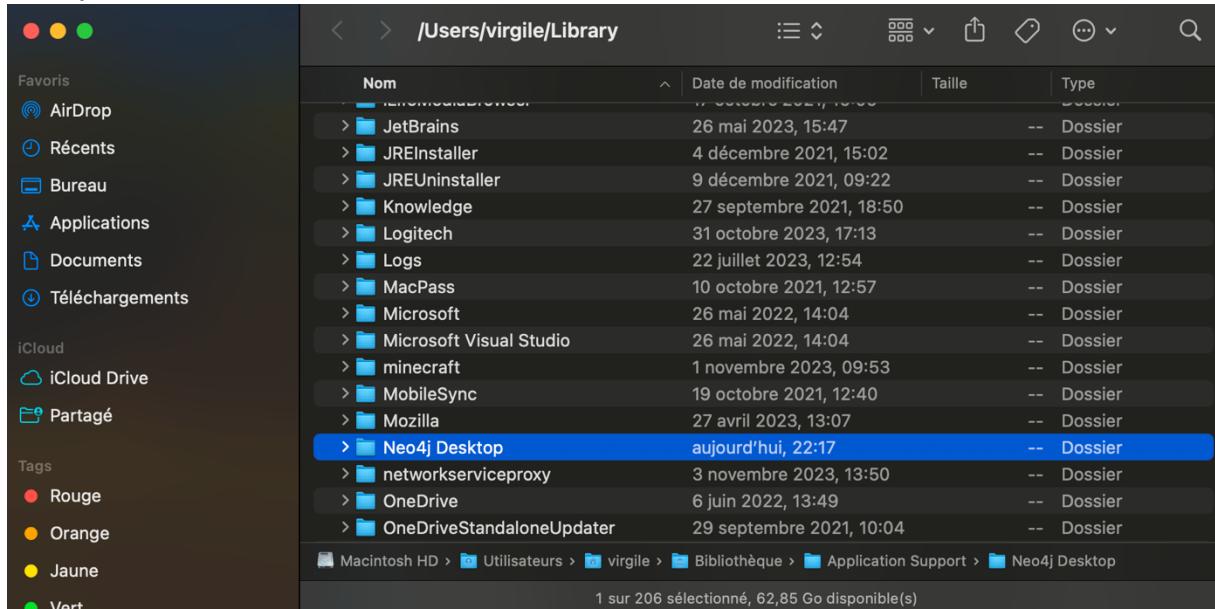
```

Une fois sur SQL nous avions une meilleure visibilité de nos données nous avons donc défini clairement les relations que nous allions mettre en place et émis quelques hypothèses quant aux requêtes que nous pourrions réaliser.

Import des fichiers

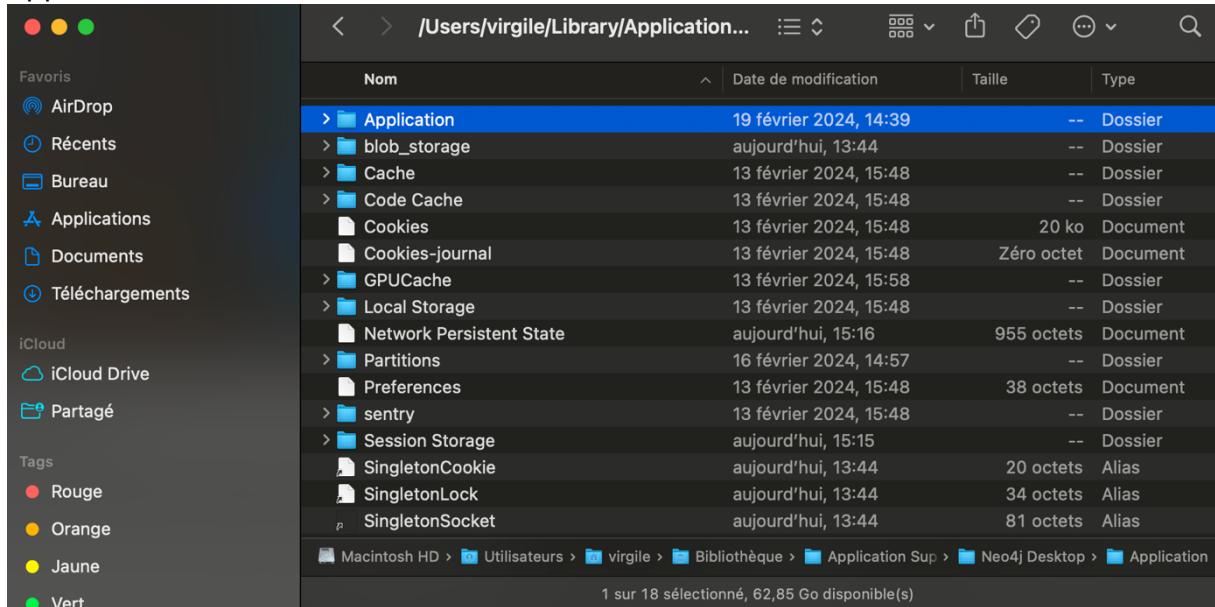
Nous avons ensuite réalisé nos importations dans neo4j. Dans un premier temps nous avons créé un projet sur neo4j et créer une base vide. Il faut ensuite exporter nos fichiers de données au format csv avant de les ranger dans le dossier adéquat sur son ordinateur : bibliothèque>Neo4j>Application>relate-data >dbmss>fichier de la base (ici dbms-884f5bf0-b01a-431a-b4d7-bb3b40dfc1c8)> import

Néo4j :



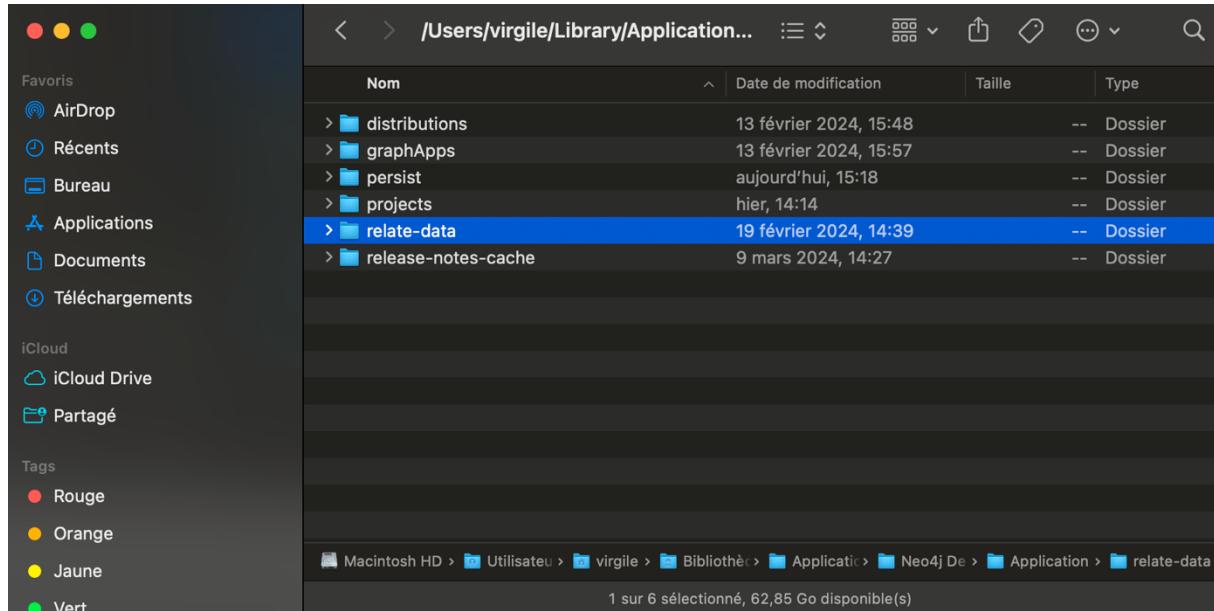
A screenshot of a Mac OS X Finder window. The path is displayed as /Users/virgile/Library. The left sidebar shows 'Favoris' with items like AirDrop, Récents, Bureau, Applications, Documents, Téléchargements, iCloud, iCloud Drive, Partagé, Tags (Rouge, Orange, Jaune, Vert), and a 'Macintosh HD' volume icon. The main pane is a file list with columns for Nom, Date de modification, Taille, and Type. The 'Neo4j Desktop' folder is highlighted with a blue selection bar. Other visible folders include JetBrains, JREInstaller, JREUninstaller, Knowledge, Logitech, Logs, MacPass, Microsoft, Microsoft Visual Studio, minecraft, MobileSync, Mozilla, networkserviceproxy, OneDrive, and OneDriveStandaloneUpdater. At the bottom of the list, there is a breadcrumb trail: Macintosh HD > Utilisateurs > virgile > Bibliothèque > Application Support > Neo4j Desktop. A status bar at the bottom indicates '1 sur 206 sélectionné, 62,85 Go disponible(s)'.

Application :

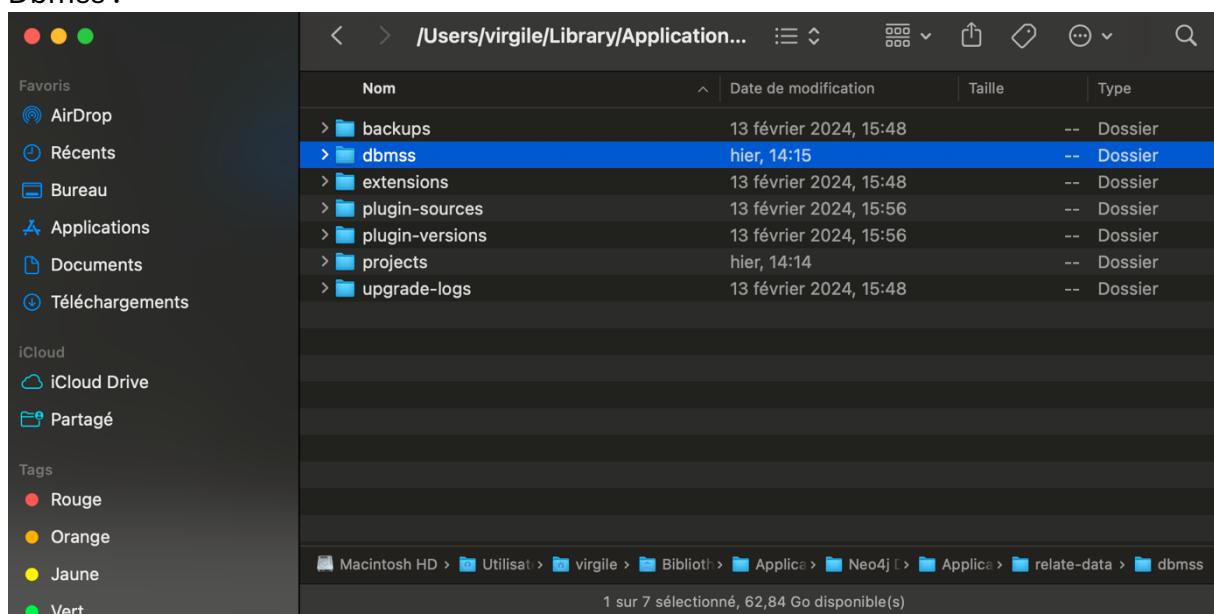


A screenshot of a Mac OS X Finder window. The path is displayed as /Users/virgile/Library/Application... (partially visible). The left sidebar is identical to the previous screenshot. The main pane shows a list of files and folders under the 'Application' folder. The 'Application' folder is selected and highlighted with a blue selection bar. Other items listed include blob_storage, Cache, Code Cache, Cookies, Cookies-journal, GPUCache, Local Storage, Network Persistent State, Partitions, Preferences, sentry, Session Storage, SingletonCookie, SingletonLock, and SingletonSocket. The status bar at the bottom indicates '1 sur 18 sélectionné, 62,85 Go disponible(s)'.

Relate-data :



Dbmss :



Fichier de notre base (dbms-884f5bf0-b01a-431a-b4d7-bb3b40dfc1c8) :

Nom	Date de modification	Taille	Type
> dbms-9fe075b0-99...89bf-b900de3e814d	21 février 2024, 18:28	--	Dossier
> dbms-63c4c5f9-92...-a085-e04ab21f5cf	13 février 2024, 15:58	--	Dossier
> dbms-281ab592-bd...a7f3-369319d0357b	13 février 2024, 17:15	--	Dossier
> dbms-884f5bf0-b01a-431a-b4d7-bb3b40dfc1c8	hier, 14:16	--	Dossier
> dbms-bf5263d0-e0b3-405e-847e-07546ee28273	2024, 15:20	--	Dossier
> dbms-bfa042ab-0e...3-8a6a-ffe1ddbc711a	19 février 2024, 14:39	--	Dossier

Import :

Nom	Date de modification	Taille	Type
> bin	hier, 14:15	--	Dossier
> certificates	hier, 14:15	--	Dossier
> conf	hier, 14:15	--	Dossier
> data	hier, 14:15	--	Dossier
> import	aujourd'hui, 15:43	--	Dossier
> labs	hier, 14:15	--	Dossier
> lib	hier, 14:15	--	Dossier
LICENSE.txt	hier, 14:15	22 ko	Format texte
> licenses	hier, 14:15	--	Dossier
LICENSES.txt	hier, 14:15	119 ko	Format texte
> logs	hier, 14:15	--	Dossier
> metrics	hier, 14:16	--	Dossier
NOTICE.txt	hier, 14:15	135 octets	Format texte
packaging_info	hier, 14:15	53 octets	Document
> plugins	hier, 14:15	--	Dossier
> products	hier, 14:15	--	Dossier

Collage des fichiers :

The Finder window displays a list of CSV files:

Nom	Date de modification	Taille	Type
Annee.csv	aujourd'hui, 15:42	225 octets	valeurs...virgul
BDD STADES EURO 2016.csv	aujourd'hui, 13:43	685 octets	valeurs...virgul
BDD_LICENCE_FFF.csv	hier, 14:14	2 ko	valeurs...virgul
data-es-2.csv	aujourd'hui, 14:43	9,7 Mo	valeurs...virgul
Departement.csv	aujourd'hui, 14:04	4 ko	valeurs...virgul
Events.csv	hier, 14:35	757 octets	valeurs...virgul
Infrastructure.csv	aujourd'hui, 15:12	9,7 Mo	valeurs...virgul
Licenciés.csv	aujourd'hui, 16:09	9 ko	valeurs...virgul
région.csv	aujourd'hui, 14:09	509 octets	valeurs...virgul
Stade_Euro_2016.csv	aujourd'hui, 15:05	3 ko	valeurs...virgul
T18F013.csv	aujourd'hui, 13:43	509 octets	valeurs...virgul

Path: Macintosh > dbms-884f5bf0-b01a-431a-b4d7-bb3b40dfc1c8 > import

11 sur 11 sélectionnés, 62,84 Go disponible(s)

Une fois ces étapes réalisées on peut se rendre sur l'interface utilisateur de Neo4j pour importer dans le logiciel les fichiers.

Interface utilisateur :

The Neo4j Browser interface shows the following sections:

- Top Bar:** neo4j\$
- Terminal:** \$:play start
- Neo4j Logo:** neo4j
- Getting started with Neo4j Browser:** Neo4j Browser user interface guide. Buttons: Get started, Open guide.
- Try Neo4j with live data:** A complete example graph that demonstrates common query patterns. Actors & movies in cross-referenced pop culture. Button: Start querying.
- Cypher basics:** Intro to Graphs with Cypher. What is a graph database? How can I query a graph?
- Bottom Terminal:** \$:server status
- Connection status:** You are connected as user neo4j to bolt://localhost:7687. Connection credentials are stored in your web browser.

Importation des tables

Importations des fichiers : vous retrouverez si dessous les codes pour importer les fichiers dans neo4j. Ces codes récupèrent les fichiers dans le dossier import via leur lien, ils séparent les colonnes via la fonction fieldterminator (à noter qu'en fonction du fichier .csv la valeur peut changer passer d'une virgule à un point-virgule). Enfin il importe chaque ligne en les renommant au préalable et en formatant correctement certaines données (e.g. longitude : toFloat(row.`Longitude (WGS84)`))

Table Events:

```
LOAD CSV WITH HEADERS FROM "file:///Events.csv" AS row
FIELDTERMINATOR ;
CREATE (:Events {
Code:row.Code,
Année:row.Annee,
Nom:row.Nom
})
```

Exemple de l'importation de la table Events :

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with four tabs: 'Graph' (selected), 'Table', 'Text', and 'Code'. The main area displays a network graph with numerous blue circular nodes, each representing an event. One node is selected and highlighted with a larger circle, showing its properties in a floating panel on the right. The properties for this selected node are:

Events
<elementId> 4:73d3bc73-7671-4886- abf4-ccd4a8a5fed2:29
<id> 29
Année 2016
Code 30
Nom Euro

The top bar of the browser shows the command: `neo4j$ MATCH (n:Events) RETURN n LIMIT 32`.

Table licenciés (appeler licence dans le logiciel) :

```
LOAD CSV WITH HEADERS FROM "file:///Licenciés.csv" AS row
FIELDTERMINATOR ';'
CREATE (:licence {
Code:row.`Code`,
Année:row.`Année`,
Région:row.`Région`,
nb_lic:row.`NbLicenciés`})
```

Exemple de l'importation de la table Licences :

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with four tabs: Graph (selected), Table, Text, and Code. The main area displays a network graph with numerous nodes represented as small circles. One node in the center is highlighted with a larger circle and has a yellow border, indicating it is selected. To the right of the graph, a panel titled "Node properties" is open, showing details for the selected node. The properties listed are:

licence
<elementId> 4:73d3bc73-7671-4886-> abf4-ccd4a8a5fed2:44
<id> 44
Année 2022
Code 13
Région Auvergne-Rhône-Alpes
nb_lic 265136

Table départements :

```
LOAD CSV WITH HEADERS FROM "file:///Departement.csv" AS row
FIELDTERMINATOR ;
CREATE (:departement{
id_reg:row.`Code`,
departement:row.`Département`,
chef_lieu:row.`Chef-lieu`,
region:row.Région
})
```

Exemple de l'importation de la table départements :

The screenshot shows the Neo4j browser interface. On the left, there are four tabs: Graph (selected), Table, Text, and Code. The main area displays a network graph where nodes represent French departments, such as Ain, Aude, Ariège, Aveyron, Bas-Rhin, Bouches-du-Rhône, Charente, Charente-Maritime, Corrèze, Corse-du-Sud, Drôme, Gard, Haute-Garonne, Hautes-Alpes, Isère, Jura, Lot, Lot-et-Garonne, Maine-et-Loire, Manche, Marne, Mayenne, Nièvre, Puy-de-Dôme, Pyrénées-Orientales, Saône-et-Loire, Tarn, Tarn-et-Garonne, Vaucluse, and Vendée. One node, 'Ain', is highlighted with a larger size and a central circular detail, indicating it is the current selection. To the right of the graph is a panel titled 'Node properties' which lists the properties for the selected node 'Ain':

departement
<elementId> 4:73d3bc73-7671-4886-> abf4-ccd4a8a5fed2:318
<id> 318
chef_lieu Bourg-en-Bresse
departeme Ain nt
id_reg 01
region Auvergne-Rhône-Alpes

Table Régions :

```
LOAD CSV WITH HEADERS FROM "file:///Région.csv" AS row
FIELDTERMINATOR ",";
CREATE (:Region {
id_reg:row.`Code`,
région:row.`Région`,
population:row.`Pop 2018`
})
```

Exemple de l'importation de la table Régions :

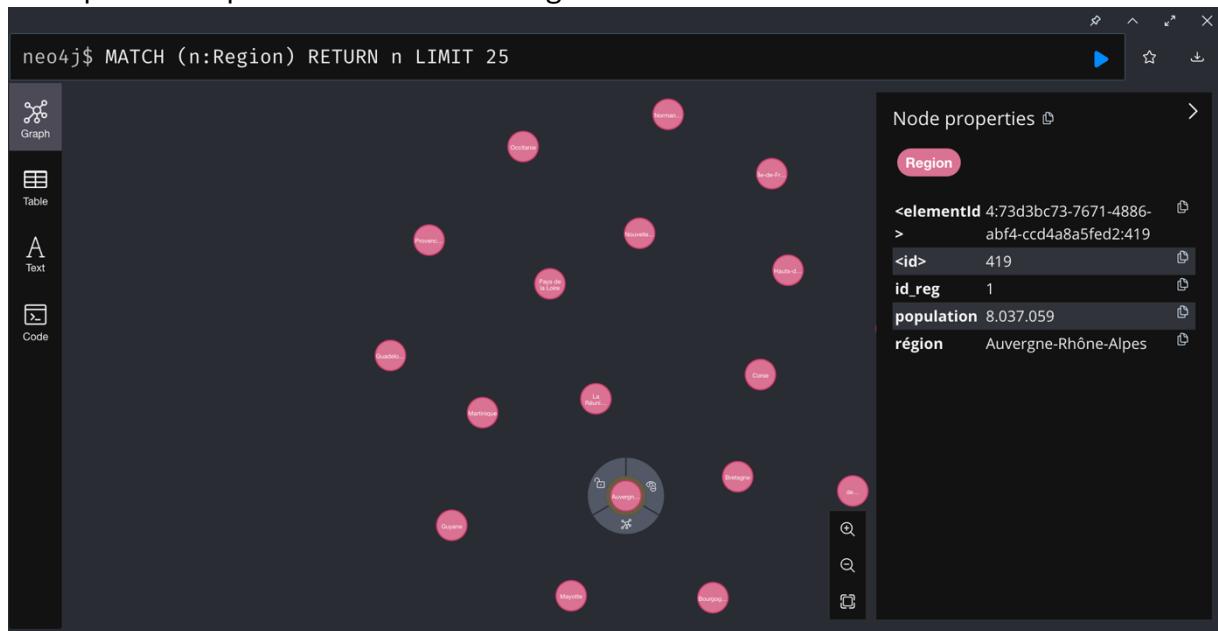


Table infrastructures renommé InstallationSportive :

```
LOAD CSV WITH HEADERS FROM "file:///Infrastructure.csv" AS row
FIELDTERMINATOR ';'
CREATE (:InstallationSportive {
    numeroInstallation: row.`Numéro de l'installation sportive`,
    nomInstallation: row.`Nom de l'installation sportive`,
    adresse: row.`Numéro, type et nom de la voie`,
    codePostal: row.`Code postal`,
    nomCommune: row.`Commune Nom`,
    inseeCommune: row.`Commune INSEE`,
    accesTransportsCommuns: row.`Accessibilité de l'installation en transport en commun`,
    installationParticuliere: row.`Installation particulière`,
    particulariteInstallation: row.`Type de particularité de l'installation`,
    accesDiffModesTransportsCommuns: row.`Accessibilité de l'installation en transport en commun des différents mode`,
    numeroEquipement: row.`Numéro de l'équipement sportif`,
    nomEquipement: row.`Nom de l'équipement sportif`,
    typeEquipement: row.`Type d'équipement sportif`,
    longitude:toFloat(row.`Longitude (WGS84)`),
    latitude:toFloat(row.`Latitude (WGS84)`),
    anneeMiseEnService: toInteger(row.`Année de mise en service`),
    typeSol: row.`Nature du sol`,
    natureEquipement: row.`Nature de l'équipement sportif`,
    periodeMiseEnService: row.`Période de mise en service`,
    codeDepartement: row.`Département Code`,
    codeRegion: row.`Région Code`,
    coordonnees: point({longitude: toFloat(row.`Longitude (WGS84)`), latitude: toFloat(row.`Latitude (WGS84)`)}),
    region:row.`Région`
})
```

Exemple de l'importation de la table InstallationSportive:

The screenshot shows the Neo4j browser interface with the following details:

- Query Bar:** The query `neo4j$ MATCH (n:InstallationSportive) RETURN n LIMIT 25` is entered.
- Graph View:** A visualization of the imported data as a network of nodes. Most nodes are small orange circles, while one specific node in the center is significantly larger and contains a small tree icon, representing the root node of the imported data.
- Properties Sidebar:** The right-hand sidebar displays the properties of the selected node, which are:
 - elementId:** 4:73d3bc73-7671-4886-abf4-ccd4a8a5fed2:437
 - <id>:** 437
 - accesDiff:** {}
 - ModesTra:** nsportsCo
 - nsportsCo:** mmuns
 - sportsCom:** muns
 - accesTrans:** true
 - EnService:** anneeMiseEnService
 - departement:** codeDepartement
 - region:** codeRegion
 - adresse:** Chemin de Recourt
 - codePostal:** 62156

Table des stades de l'euro 2016 :

```
LOAD CSV WITH HEADERS FROM "file:///Stade_Euro_2016.csv" AS row
FIELDTERMINATOR ';
CREATE (:Stades {
ID_SE:row.`ID_Stade_Event`,
Nom_Stade:row.`Nom_Stade`,
Ville_Stade:row.`Ville_Stade`,
Capacite_Stade:row.`Capacite_Stade`,
Region_Stade:row.`Region_Stade`,
Année_Construction:row.`Année_Construction`,
Code_evenement:row.`Code_evenement`,
ID_Stade:row.`ID_Stade`})
```

Exemple de l'importation de la table Stade euro 2016 :

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with tabs for Graph, Table, Text, and Code. The Graph tab is selected, displaying a network graph with several nodes represented by grey circles of varying sizes. One node in the center is highlighted with a larger circle and a yellow border, indicating it is the current focus. To the right of the graph, the Node properties panel is open, showing detailed information for this selected node. The properties are listed in a table:

Stades
<element> 4:73d3bc73-7671-4886-abf4-ccd4a8a5fed2:38027
<id> 38027
Année_Construction 1972
Capacite_Stade 45000
Code_evenement 8
ID_SE 19
ID_Stade 5
Nom_Stade Parc des Princes
Region_Stade Île-de-France

Table Année :

```
LOAD CSV WITH HEADERS FROM "file:///Annee.csv" AS row
FIELDTERMINATOR ';
CREATE (:Année {
id_annee:row.ID,
Année:row.Année
})
```

Exemple de l'importation de la table Année :

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with four tabs: Graph (selected), Table, Text, and Code. The main area displays a network graph where nodes represent years, with their labels (e.g., 1900, 1950, 2000) visible on or near the nodes. A specific node for the year 2000 is selected, highlighted with a larger size and a yellow border. To the right of the graph, the 'Node properties' panel is open, showing the following details for the selected node:

Année
<elementId> 4:1ea8fefd-dbdd-4fc5-98d4-9ccb14befbb4:150761
<id> 150761
Année 2000
id_annee 1

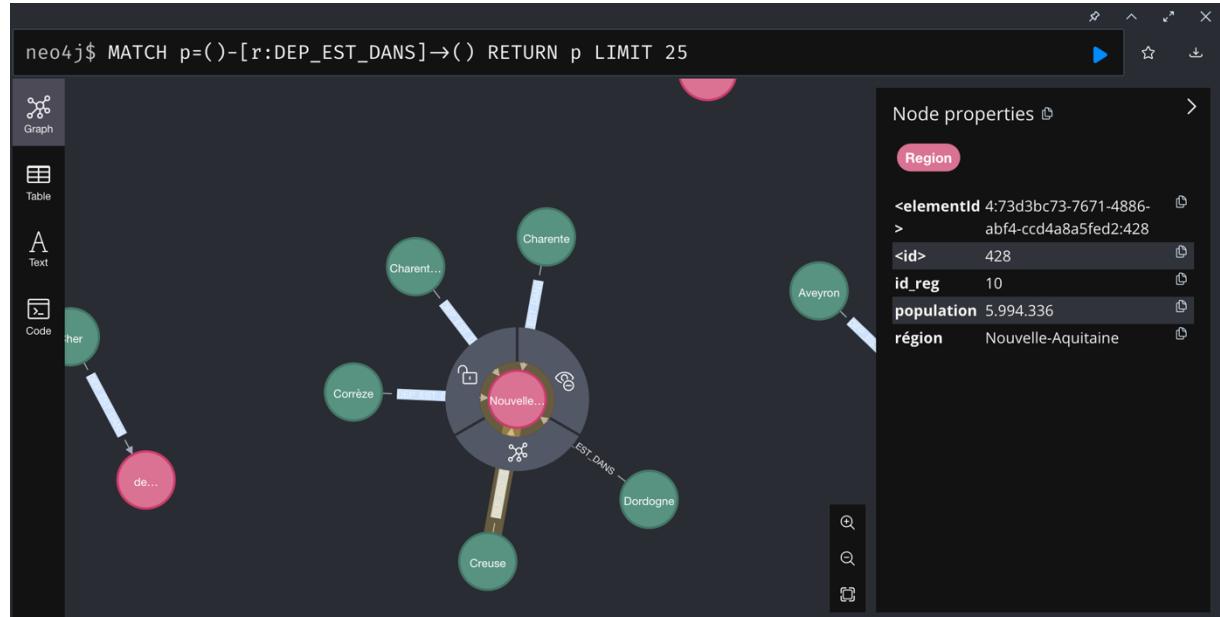
Création de relations

A ce stade la du processus toutes les données sont importantes cependant aucune ne sont connectés entre elles c'est pourquoi les codes suivants auront pour but de connecter les tables entre elles.

Ce code va créer une relation entre les départements et les régions en les connectant grâce au nom de la région. Si le nom est mal orthographié (en vue d'une mise à jour) la relation ne sera pas faite entre les points.

```
MATCH (d:departement), (r:Region)
WHERE d.region = r.région
CREATE (d)-[:DEP_EST_DANS]->(r)
```

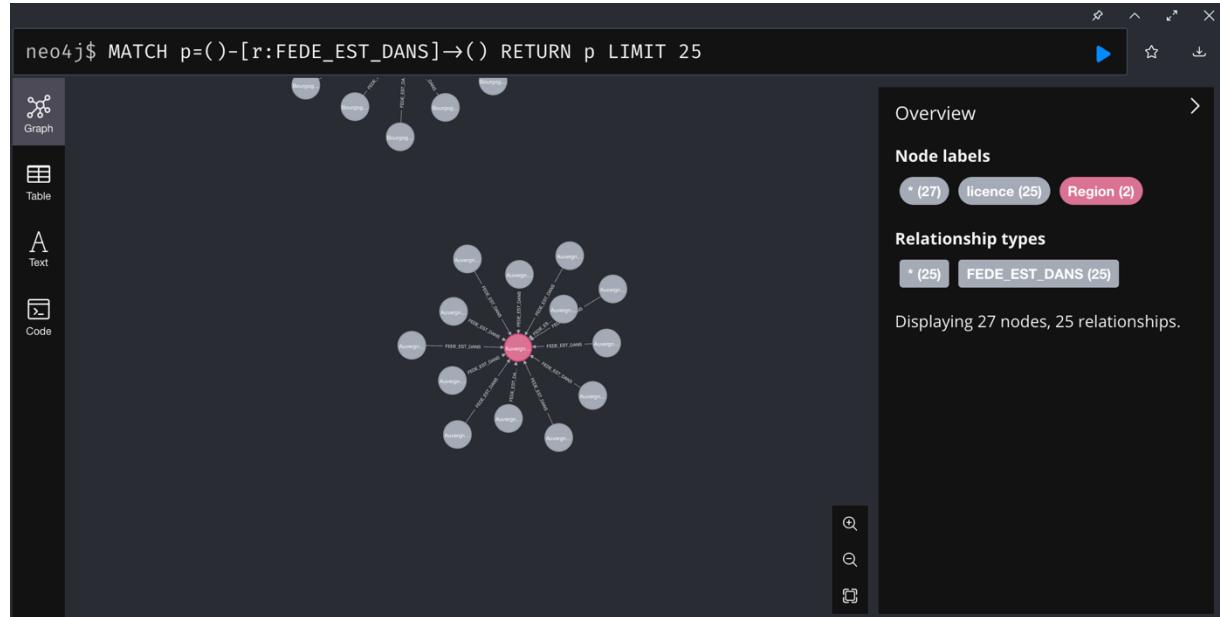
Cet exemple montre le lien entre la Nouvelle-Aquitaine et ses départements, la relation est notée « EST_DANS »



Ce code va créer une relation entre les fédérations et les régions en les connectant grâce au nom de la région. Si le nom est mal orthographié (en vue d'une mise à jour) la relation ne sera pas faite entre les points.

```
MATCH (l:licence), (r:Region)
WHERE l.Région = r.région
CREATE (l)-[:FEDE_EST_DANS]->(r)
```

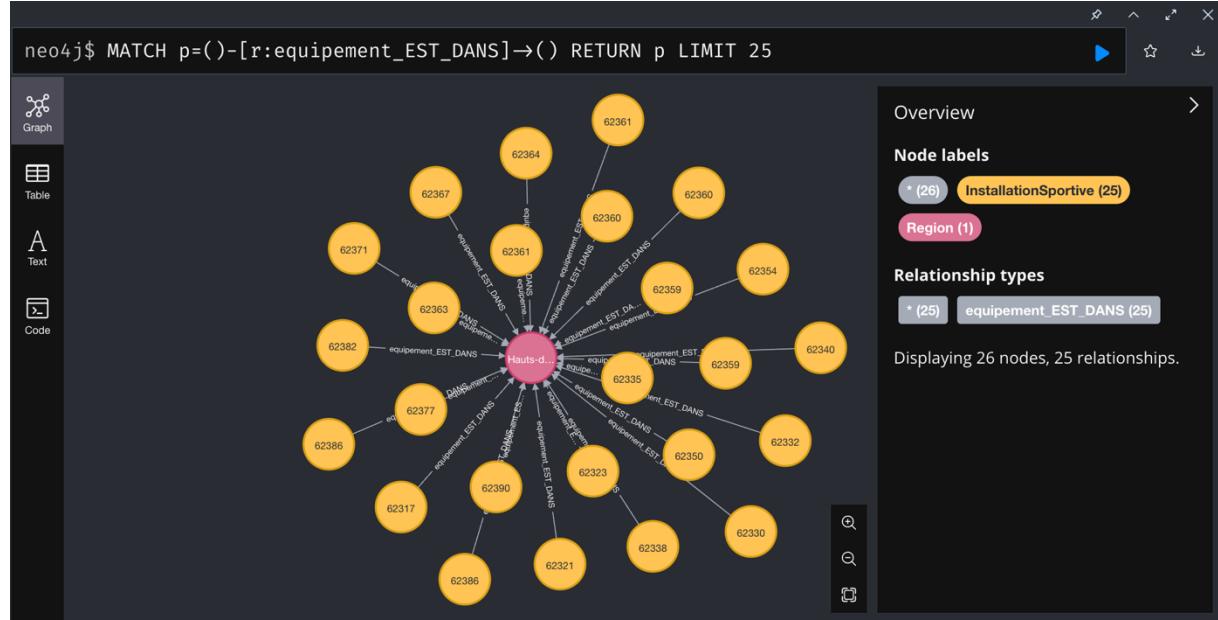
On constate que les bulles grises correspondent à une fédération par année elles sont toutes connectées à leur lieu



Ce code va créer une relation entre les Installations sportives et les régions en les connectant grâce au nom de la région. Si le nom est mal orthographié (en vue d'une mise à jour) la relation ne sera pas faite entre les points.

```
MATCH (i:InstallationSportive), (r:Region)
WHERE i.region = r.région
CREATE (i)-[:equipement_EST_DANS]->(r)
```

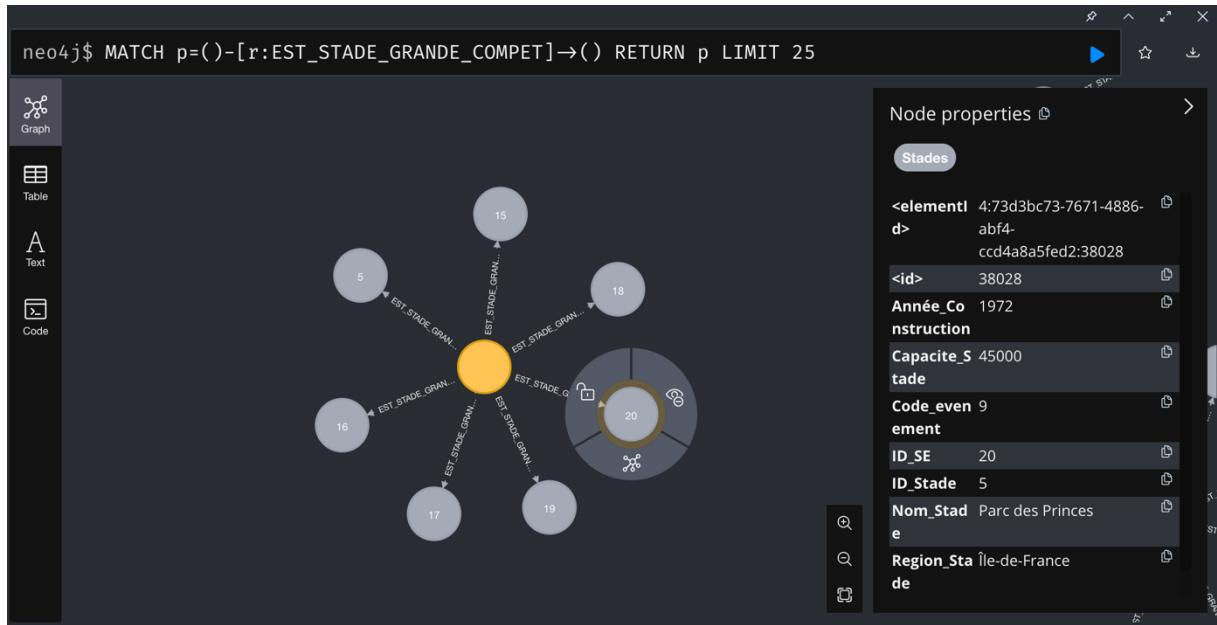
Voici un échantillon d'équipement situer dans les hauts de France, chaque correspond à un équipement nommé par son ID. les relations sont nommés « equipement_EST_DANS »



Ce code va créer une relation entre les Installations sportives et les stades de l'euro en les connectant grâce à l'id de l'installation.

```
MATCH (i:InstallationSportive), (s:Stades)
WHERE i.numerolInstallation = s.ID_Stade
CREATE (i)-[:EST_STADE_GRANDE_COMPET]->(s)
```

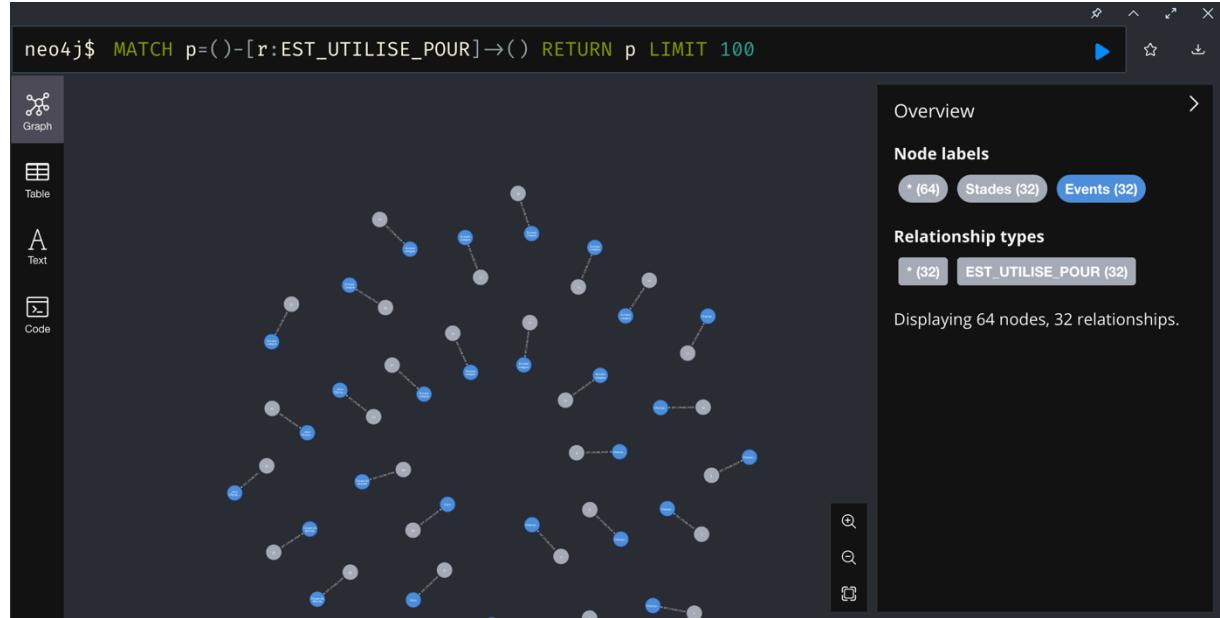
L'exemple si dessous relie les grandes installations parmi toutes les installations:



Cette commande permet de lier les grands stades et l'évènement pour lequel il a été utilisé

```
MATCH (s:Stades), (r:Events)  
WHERE s.ID_SE = r.Code  
CREATE (s)-[:EST_UTILISE_POUR]->(r)
```

Exemple des liens, ils sont modélisés par le lien « EST_UTILISE_POUR » :



Cette dernière relation nous permet de relier les évènements en fonction de leur date

```
MATCH (s:Annee), (r:Events)  
WHERE s.Annee = r.Annee  
CREATE (r)-[:EST_EN]->(s)
```

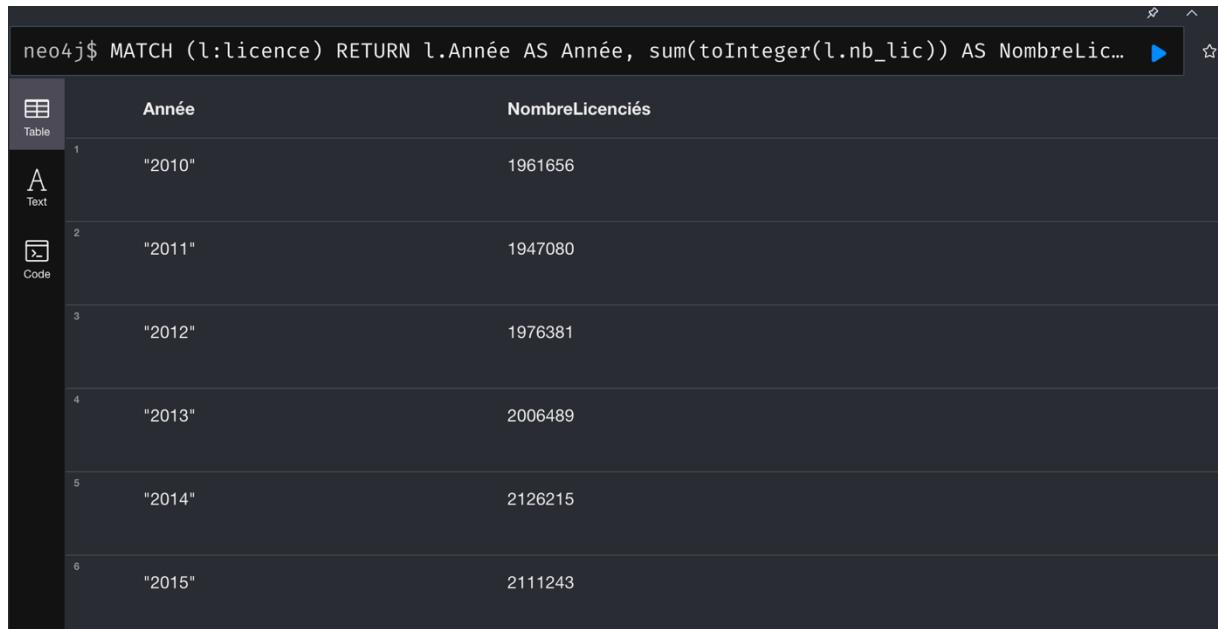
Après avoir importé l'ensemble de nos nœud et relation nous arrivons à un total de 38050 nœuds et 36081 relations.

Nous allons donc exécuter quelques requêtés pour répondre à notre problématique. Dans le cadre d'une étude statistique il faudrait augmenter le nombre de requêtes et pousser l'exercice d'analyse. Nous avons créé un modèle pour exploiter des données via neo4j. Ces quelques requêtes permettent de tirer des résultats partiels.

Requête de visualisation

Cette requête nous permet de voir le nombre de licencié par année

```
MATCH (l:licence)
RETURN l.Année AS Année, sum(toInteger(l.nb_lic)) AS NombreLicenciés
ORDER BY l.Année
```



	Année	NombreLicenciés
1	"2010"	1961656
2	"2011"	1947080
3	"2012"	1976381
4	"2013"	2006489
5	"2014"	2126215
6	"2015"	2111243

Cette requête nous permet de calculer le nombre de licenciés dans la région Île de France par année

```
MATCH (l:licence)
WHERE l.Région = 'Île-de-France'
RETURN l.Année AS Année, sum(toInteger(l.nb_lic)) AS NombreLicenciés
ORDER BY l.Année
```

The screenshot shows the Neo4j browser interface with a query results table. The table has two columns: 'Année' (Year) and 'NombreLicenciés' (Number of Licensees). The data rows are numbered 1 through 6, corresponding to the years 2010 through 2015 respectively. The table is displayed in a dark-themed browser window.

	Année	NombreLicenciés
1	"2010"	221171
2	"2011"	224935
3	"2012"	231118
4	"2013"	240431
5	"2014"	264885
6	"2015"	261654

Started streaming 13 records in less than 1 ms and completed after 2 ms.

Requête en réponse à la problématique

La requête suivante consiste à calculer la différence du nombre de licencié d'une année à l'autre :

```
MATCH (l:licence)
WITH l.Année AS Année, sum(toInteger(l.nb_lic)) AS NombreLicenciés
ORDER BY Année
WITH collect({année: Année, licenciés: NombreLicenciés}) AS data
UNWIND range(1, size(data) - 1) AS idx
WITH data[idx].année AS Année, data[idx].licenciés - data[idx-1].licenciés AS DiffLicenciés
RETURN Année, DiffLicenciés
ORDER BY Année
```

On constate qu'il y a certaines années charnières dans l'augmentation du nombre de licences footballistique en France avec de net augmentation en 2014 pouvant être expliqué par un bon parcours en coupe du monde de l'équipe de France, une augmentation en 2016 pour le bon parcours et le fait que la compétition soit en France enfin en 2018 lorsque la France est championne du monde. Ces données nous permettent également de constater l'impact de la crise COVID-19 sur les licences des joueurs de foot.

	Année	DiffLicenciés
2	"2012"	29301
3	"2013"	30108
4	"2014"	119726
5	"2015"	-14972
6	"2016"	48922
7	"2017"	-20073
8	"2018"	58732
9	"2019"	-74489
10	"2020"	-222386
11	"2021"	228105
12	"2022"	76698

La requête suivante nous permet d'observer le nombre de nouvelles Installations Sportives via la variable « Année de mise en service »

```
MATCH (i:InstallationSportive)
WHERE i.anneeMiseEnService IN [2010,2011,2012,2013,2014,2015, 2016,
2017,2018,2019,2020,2021,2022,2023,2024]
RETURN i.anneeMiseEnService AS Année, count(i) AS NombreInfrastructuresCrées
ORDER BY Année
```

Les résultats nous permettent de constater que le nombre de nouvelle installation diminue drastiquement au fil des ans. Néanmoins cette théorie nous paraît en désaccord avec les résultats obtenus précédemment. Nous avons constaté qu'un grand nombre de ligne n'avait pas d'information de mise en service renseigner. Nous supposons que la mauvaise mise à jour du fichier ne nous permet pas de tirer de conclusion qu'il faudrait donc croiser ces résultats avec d'autres études pour valider ou réfuter ce résultat

	Année	NombreInfrastructuresCrées
1	2010	309
2	2011	203
3	2012	199
4	2013	144
5	2014	127
6	2015	93
7	2016	61
8	2017	26
9	2018	27
10	2019	33
11	2020	13

Conclusion:

En conclusion nous pouvons dire que les évènements d'importance mondial ont un réel impact sur le football Français à travers ses licenciés. Il serait néanmoins intéressant de creuser davantage ces résultats.