TP Stations et Trains (RER et métro) en Ile de France

Antonin Durand

INFI 3

Table des matières

1- Conception de la base de données

Dans un premier temps, on réfléchit à une conception de bases de données permettant de stocker les différentes lignes, arrêts et bus, dans le but de répondre aux questions de la partie 1.

a) Représentation graphique de la base de données de manière relationnelle

Ci-dessous un graphique représentant la base de données de manière relationnelle

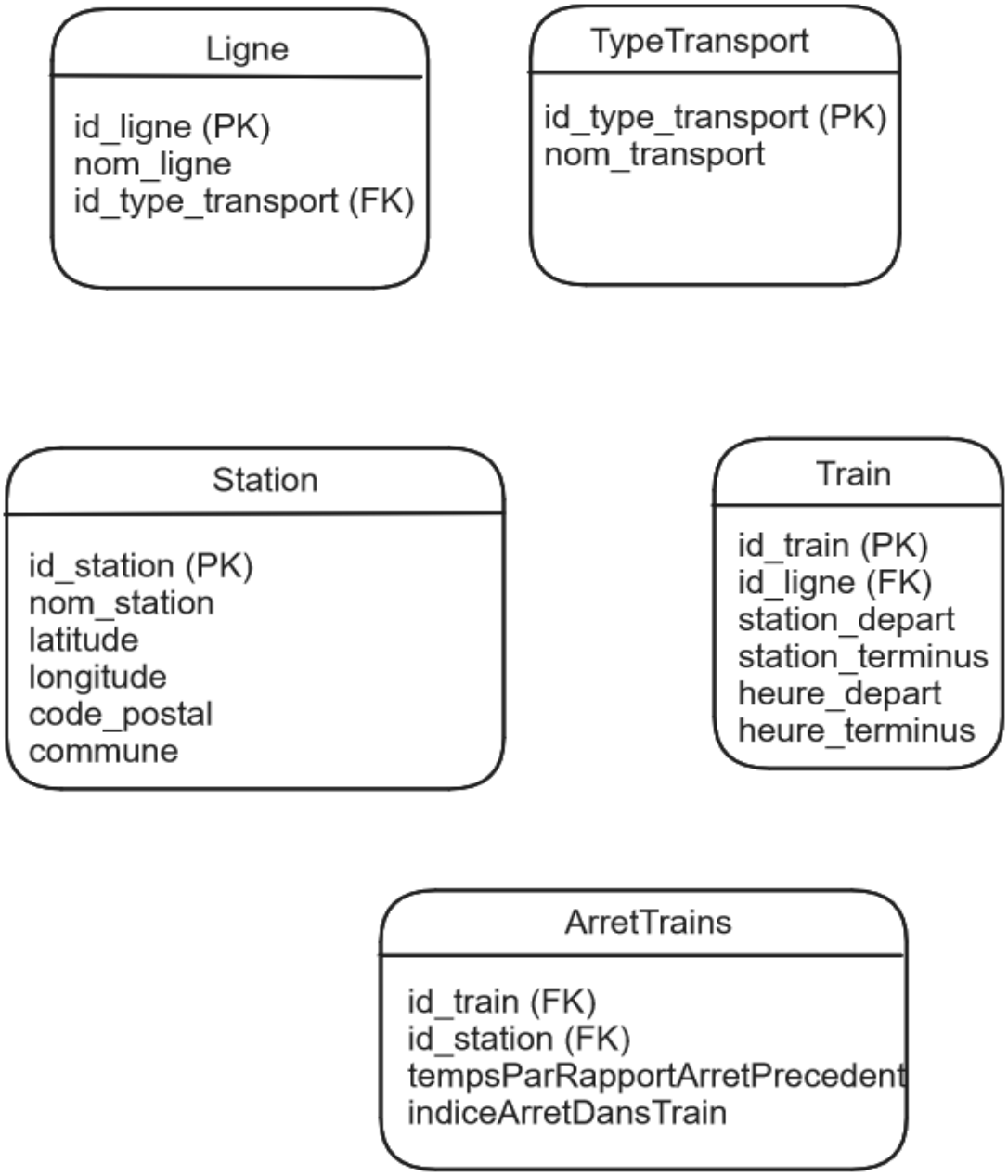

Illustration 1: Schéma relationnel de la base de données des stations et trains

PK : primary key
FK : foreing key

b) Description des tables

La table **Ligne** contient les informations de chaque ligne comme le nom de la ligne (C, N, ...) ainsi que le transport utilisé.

La table **TypeTransport** permet de lister tous les types de transport dans la base de données. Cela a pour objectif de pouvoir identifier rapidement tous les transports présents (RER, Métro, Train, …) dans la base de données, et de pouvoir les normer.

La table **Station** contient les informations de toutes les stations ou arrêts présents dans le réseau de transport en Île de France. Elle contient par exemple le nom de la station, ses coordonnées géogaphiques, la commune dans laquelle la station se trouve, ainsi que le code postal associé.

La table **Train** permet de répondre à la problématique qui est qu'une ligne peut avoir plusieurs arrêts de départs, et plus terminus, comme la ligne C. Ainsi, il faut créer une table, qu'on nomme Train, et qui permet de stocker les informations de chaque train (RER, métro, …), comme la ligne dans lequel il circule, la station de départ et d'arrivée.

La table **ArretTrains** permet de stocker les arrêts dans lesquels chaque train desserve. Est stocké le train (train 1 de la ligne C), l'identifiant de la station, sa position dans le parcours de la ligne, ainsi que le temps en minutes mis par le train pour arriver à l'arrêt actuel depuis l'arrêt précédent.

https://data.iledefrance-mobilites.fr/explore/dataset/traces-des-lignes-de-transport-en-commun-idfm/table/?disjunctive.route_type

C) Script création de la base de données

```
CREATE TABLE TypeTransport(
    id_type_transport INT PRIMARY KEY,
    nom_transport VARCHAR(100) not null
)
CREATE TABLE Station(
    id_station INT PRIMARY KEY NOT NULL,
    nom_station VARCHAR(100) NOT NULL,
    latitude FLOAT NOT NULL,
    longitude FLOAT NOT NULL,
    commune VARCHAR(100) NOT NULL,
    code_postal INT NOT NULL
);
CREATE TABLE Ligne(
    id_ligne INT PRIMARY KEY NOT NULL,
    nom_ligne VARCHAR(100) NOT NULL,
    id_type_transport INT NOT NULL,
    foreign key (id_type_transport) REFERENCES TypeTransport(id_type_transport)
);
CREATE TABLE Train(
    id_train INT PRIMARY KEY NOT NULL,
    id_ligne INT NOT NULL,
    station_depart INT NOT NULL,
    station_terminus INT NOT NULL,
    heure_depart TIME NOT NULL,
    heure_terminus TIME NOT NULL,
    FOREIGN KEY (id_ligne) REFERENCES Ligne(id_ligne),
    FOREIGN KEY (station_depart) REFERENCES Station(id_station),
```

```
    FOREIGN KEY (station_terminus) REFERENCES Station(id_station)
);
CREATE TABLE ArretTrains(
    id_train INT NOT NULL,
    id_station INT NOT NULL,
    tempsParRapportArretPrecedent int NOT NULL,
    indiceArretDansTrain INT NOT NULL,
    FOREIGN KEY train(id_train) REFERENCES Train(id_train) ,
    FOREIGN KEY station(id_station) REFERENCES Station(id_station)
);
```

2 – Requetes

Dans cette partie, nous allons exprimer les requêtes SQL pour répondre à chaque question.

1. Etant donnée une ligne X (e.g. RER B), quelles sont ses stations?

```
select distinct l.id_ligne, l.nom_ligne, st.id_station, st.nom_station,
st.latitude, st.code_postal, st.commune
from Ligne l, Station st, Train, tr, ArretTrains at
where l.id_ligne = tr.id_ligne
    and at.id_train = tr.id_train
    and st.id_station = tr.id_train
    and l.nom_ligne = "X";
```

2. Quelles sont toutes les stations au sud (géographiquement) d'une station X donnée?

```
select distinct st.id_station, st.nom_station, st.latitude, st.longitude,
st.code_postal, st.commune
from Station st
where st.latitude < X_latitude;
```

3. Quelles sont toutes les stations au sud d'une station X donn´ee et qui sont desservies par le même train?

```
select distinct st.id_station, st.nom_station, st.latitude, st.longitude,
st.code_postal, st.commune, at.id_train, at.tempsParRapportArretPrecedent,
at.indiceArretDansTrain
from Station st, ArretTrains at
where st.id_station = at.id_train
    and st.latitude < X_latitude
    and at.id_train = X.id_train;
```

4. Etant donnée deux stations X et Y , y-a-t-il une connexion directe de X à Y (un train qui va de X à Y)? (question pas triviale, prendre comme exemple les stations Robinson et Massy-Palaiseau).

```
SELECT COUNT(*) AS ConnexionDirecte
FROM (
        SELECT DISTINCT T1.id_train, T1.heure_depart AS DepartRobinson,
T2.heure_arrivee AS ArriveeMassyPalaiseau
        FROM Train AS T1
                JOIN Train AS T2 ON T1.id_train = T2.id_train
                JOIN Station AS S1 ON T1.station_depart = S1.id_station
                JOIN Station AS S2 ON T2.station_terminus = S2.id_station
        WHERE S1.nom_station = 'Robinson' AND S2.nom_station = 'Massy-
Palaiseau'
          AND T1.heure_depart < T2.heure_arrivee
    ) AS Connexions;
```

5. Etant donnée deux stations X et Y de la même ligne, combien de stations les séparent? (attention: requête imprécise)

```
SELECT COUNT(*) - 1 AS NombreDeStationsEntre
FROM ArretTrains AS A1
         JOIN ArretTrains AS A2 ON A1.id_train = A2.id_train
WHERE A1.id_station = (SELECT id_station FROM Station WHERE nom_station =
'StationX')
  AND A2.id_station = (SELECT id_station FROM Station WHERE nom_station =
'StationY');
```

6. Etant donnée deux stations X et Y pour lesquelles il existe une connexion directe, quelle est la durée minimale d'un trajet de X à Y ?

```
SELECT MIN(TIMEDIFF(T2.heure_arrivee, T1.heure_depart)) AS DureeMinimale
FROM Train AS T1
         JOIN Train AS T2 ON T1.id_train = T2.id_train
         JOIN Station AS S1 ON T1.station_depart = S1.id_station
         JOIN Station AS S2 ON T2.station_terminus = S2.id_station
WHERE S1.nom_station = 'StationX' AND S2.nom_station = 'StationY'
  AND T1.heure_depart < T2.heure_arrivee;
```

7. Etant donnée deux stations X et Y pour lesquelles il n'existe pas une connexion directe, y-a-t-il une connexion avec un seul changement entre les deux stations?

```
WITH ConnexionsIntermediaires AS (
    SELECT DISTINCT A1.id_station AS StationIntermediaire
    FROM ArretTrains AS A1
    JOIN ArretTrains AS A2 ON A1.id_train = A2.id_train
    WHERE A1.id_station <> A2.id_station
    AND A1.id_station IN (SELECT id_station FROM Station WHERE nom_station =
'StationX')
    AND A2.id_station IN (SELECT id_station FROM Station WHERE nom_station =
'StationY')
    )
SELECT COUNT(*) AS NombreDeConnexionsAvecChangement
FROM ConnexionsIntermediaires;
```

8. Combien de changements de ligne pour aller de la station X à la station Y ?

```
select *
from temp_liste_arrets_desservis_par_train
where
WITH Itineraires AS (
    SELECT DISTINCT
    A1.id_train AS TrainDepart,
    A2.id_train AS TrainArrivee,
    L1.nom_ligne AS LigneDepart,
    L2.nom_ligne AS LigneArrivee
    FROM
    ArretTrains AS A1
    JOIN ArretTrains AS A2 ON A1.id_train = A2.id_train
    JOIN Ligne AS L1 ON A1.id_train = L1.id_ligne
    JOIN Ligne AS L2 ON A2.id_train = L2.id_ligne
```

```
    WHERE
    A1.id_station IN (SELECT id_station FROM Station WHERE nom_station =
'StationX')
    AND A2.id_station IN (SELECT id_station FROM Station WHERE nom_station =
'StationY')
    AND A1.tempsParRapportArretPrecedent < A2.tempsParRapportArretPrecedent
    )
SELECT
        COUNT(DISTINCT TrainDepart) - 1 AS NombreDeChangementsDeLigne
FROM
    Itineraires;
```

9. Quel est le nombre maximal de changements pour un trajet entre deux stations? (pire de cas)

10. Quelles sont les stations X et Y dont le trajet le plus simple demande un nombre de changements maximal?

Les questions 9 et 10 ne peuvent pas être exprimées sous forme de requête SQL.

3 – Limites de cette solution

La cloture transitive, relation parent, enfant

On constate une première limite de cette structure en SQL, où la façon d'enregistrer les données compliquent énormément les requêtes, et peut rendre impossible d'autre fonctionnalités, comme la recherche d'un itinéraire le plus court.

Autre cas de figure, on souhaite connaître les 5 arrêts qui en précédent un autre, on peut le faire car c'est borné. Cependant, si on souhaite voir les n arrêts, on ne peut pas car ce n'est pas borné, et qu'on ne peut pas effectuer de requête récursive en sql. Une des solutions en java, python par exemple, serait de créer une liste chaînée, où chaque arrêt d'un train possède un arrêt précédent et un arrêt suivant.

4 – Script SQL

Je n'ai pas eu le temps de créer le script pour insérer des données dans la base de données. Néanmoins, le lien suivant https://data.iledefrance-mobilites.fr/explore/dataset/emplacement-des-gares-idf/table/?q=C01377&location=20,48.84368,2.3247&basemap=jawg.streets, contient un fichier csv avec la liste des lignes et stations desservies.

Il faudrait créer un script en PHP où il faut mapper les données contenues dans le fichier csv, pour pouvoir les insérer dans la base de données décrite précédemment.