

# Stations et Trains (RER et métro) en île de France

<b>Code couleur :</b>	<b>1</b>
<b>Proposition des tables :</b>	<b>1</b>
<b>Requêtes :</b>	<b>2</b>
<b>Données :</b>	<b>5</b>
Demande 1 (script de la table) :	5

## Code couleur :

- primary key
- Clé étrangère

## Proposition des tables :

1. **Table "Stations" :**
  - "IdStation" : Identifiant unique de la station.
  - "NomStation" : Nom de la station.
  - "LatitudeStation" : Coordonnée géographique : Latitude de la station.
  - "LongitudeStation" : La longitude de la station.

Table stations qui stock les stations.

2. **Table "Lignes" :**
  - "IdLignes" : Identifiant unique de la ligne.
  - "NomLignes" : Nom de la ligne (e.x : RER B)

Table ligne qui stock les lignes du réseaux.

3. **Table "Trains" :**
  - "IdTrains" : Identifiant unique du train.
  - "NomTrains" : Nom ou numéro du train (e.x : Train A).
  - "Ligne\_ID" : Clé étrangère faisant référence à l'ID de la ligne sur laquelle ce train circule.
  - "Heure\_Depart" : Heure de départ du train.
  - "Heure\_Arrivee" : Heure d'arrivée du train.

Table trains qui stock les données des trains et tous les trains.

4. **Table "Passage"** : (pour faire la liaison entre le train et la station car un train appartient à une ligne mais il passe pas par toutes les stations de la ligne)
- "Station\_ID" : Clé étrangère : l'ID de la station.
  - "Train\_ID" : Clé étrangère : l'ID du train.
  - "Heure\_Passage" : Heure de passage du train

La table Passage va stocker les données des trains passer par un gare.

5. **Table "Stations\_Lignes"** (table de liaison entre les stations et les lignes ce qui permet de savoir les stations qui dessert sur la ligne. Il est nécessaire de faire cette relation car un train ne passe pas forcément dans une station alors que la station appartient à la ligne) :
- "Station\_ID" : Clé étrangère : l'ID de la station.
  - "Ligne\_ID" : Clé étrangère : l'ID de la ligne.
  - "Direction" : Indique la direction dans laquelle le train circule sur cette ligne à partir de cette station (nord, sud, est, ouest, etc.).

La table stations lignes va stocker les gares qui appartiennent à une ligne.

6. **Table "Connexions"** (pour représenter les connexions directes entre les stations) :
- "Station\_A\_ID" : Clé étrangère : l'ID de la station de départ.
  - "Station\_B\_ID" : Clé étrangère : l'ID de la station d'arrivée.
  - "Durée\_Minimale" : Durée minimale du trajet entre les deux stations.

La table connexion va stocker les connexions des gares.

## Requêtes :

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

```
SELECT * FROM Stations
JOIN Stations_Lignes ON Stations.IdStation = Stations_Lignes.Station_ID
JOIN Lignes ON Stations_Lignes.Ligne_ID = Lignes.IdLignes
WHERE Lignes.NomLignes = 'RER B';
```

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

```
SELECT * FROM Stations
WHERE LatitudeStation < (SELECT LatitudeStation
                        FROM Stations
                        WHERE NomStation = 'Nom de la station X')
AND NomStation != 'Nom de la station X';
```

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

```
SELECT DISTINCT S.*
FROM Stations AS S
INNER JOIN Passage AS P1 ON S.IdStation = P1.Station_ID
INNER JOIN Passage AS P2 ON P1.Train_ID = P2.Train_ID
WHERE S.LatitudeStation < (SELECT LatitudeStation
                        FROM Stations
                        WHERE NomStation = 'Nom de la station X')
AND S.NomStation != 'Nom de la station X';
```

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 Nombre_de_Connexions_Directes
FROM Connexions
WHERE (Station_A_ID = ID_de_la_station_X
      AND Station_B_ID = ID_de_la_station_Y)
      OR (Station_A_ID = ID_de_la_station_Y
      AND Station_B_ID = ID_de_la_station_X);
```

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

```
SELECT COUNT(*) - 1 AS Stations_Séparées
FROM Stations_Lignes AS SL1
JOIN Stations_Lignes AS SL2 ON SL1.Ligne_ID = SL2.Ligne_ID
WHERE SL1.Station_ID = ID_de_la_station_X
      AND SL2.Station_ID = ID_de_la_station_Y
      AND SL1.Direction = SL2.Direction;
```

Ici count(\*) permet de compter le nombre de station les séparants, j'ai retiré -1 car la stations d'arrivée est compté alors que ce n'est pas une station de plus qui sépare.

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 Durée_Minimale
FROM Connexions
WHERE (Station_A_ID = ID_de_la_station_X
      AND Station_B_ID = ID_de_la_station_Y)
OR (Station_A_ID = ID_de_la_station_Y
    AND Station_B_ID = ID_de_la_station_X);
```

Ici un "or" est mis car il y a l'aller et le retour

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 ?

```
SELECT COUNT(*) AS Nombre_de_Correspondances
FROM Connexions AS C1
INNER JOIN Connexions AS C2 ON C1.Station_B_ID = C2.Station_A_ID
WHERE C1.Station_A_ID = ID_de_la_station_X
      AND C2.Station_B_ID = ID_de_la_station_Y;
```

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

```
SELECT COUNT(*) AS Nombre_de_Changements_de_Ligne
FROM ( SELECT DISTINCT Ligne_ID
      FROM Stations_Lignes
      WHERE Station_ID = ID_de_la_station_X
            OR Station_ID = ID_de_la_station_Y ) AS lignes_communes
WHERE Ligne_ID NOT IN ( SELECT Ligne_ID
                      FROM Stations_Lignes
                      WHERE Station_ID = ID_de_la_station_X
                            OR Station_ID = ID_de_la_station_Y );
```

La requête fait :

1. Elle commence par trouver toutes les lignes desservant les stations X et Y en utilisant une sous-requête.
  2. Ensuite, elle filtre les lignes pour exclure celles qui sont communes aux deux stations (celles desservant à la fois X et Y).
  3. Le résultat final est le nombre de lignes restantes, ce qui représente le nombre de changements de ligne nécessaires pour aller de X à Y.
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 ?

## Données :

Demande 1 (script de la table) :

```
CREATE TABLE Stations (  
    IdStation INT PRIMARY KEY AUTO_INCREMENT,  
    NomStation VARCHAR(255) NOT NULL,  
    LatitudeStation DECIMAL(9,6) NOT NULL,  
    LongitudeStation DECIMAL(9,6) NOT NULL  
);  
  
CREATE TABLE Lignes (  
    IdLignes INT PRIMARY KEY AUTO_INCREMENT,  
    NomLignes VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE Trains (  
    IdTrains INT PRIMARY KEY AUTO_INCREMENT,  
    NomTrains VARCHAR(255) NOT NULL,  
    Ligne_ID INT,  
    Heure_Depart TIME NOT NULL,  
    Heure_Arrivee TIME NOT NULL,  
    FOREIGN KEY (Ligne_ID) REFERENCES Lignes(IdLignes)  
);  
  
CREATE TABLE Passage (  
    Passage_ID INT PRIMARY KEY AUTO_INCREMENT,  
    Station_ID INT,  
    Train_ID INT,  
    Heure_Passage TIME NOT NULL,  
    FOREIGN KEY (Station_ID) REFERENCES Stations(IdStation),  
    FOREIGN KEY (Train_ID) REFERENCES Trains(IdTrains)  
);  
  
CREATE TABLE Stations_Lignes (  
    Station_ID INT,  
    Ligne_ID INT,  
    Direction VARCHAR(255),  
    PRIMARY KEY (Station_ID, Ligne_ID),  
    FOREIGN KEY (Station_ID) REFERENCES Stations(IdStation),  
    FOREIGN KEY (Ligne_ID) REFERENCES Lignes(IdLignes)
```

);

```
CREATE TABLE Connexions (  
  Connexion_ID INT PRIMARY KEY AUTO_INCREMENT,  
  Station_A_ID INT,  
  Station_B_ID INT,  
  Durée_Minimale INT,  
  FOREIGN KEY (Station_A_ID) REFERENCES Stations(IdStation),  
  FOREIGN KEY (Station_B_ID) REFERENCES Stations(IdStation)  
);
```