

TP : SQL / python / HTML / CSS (Données SNCF)

La base de données *SNCF* fournie sur [moodle](#) représente les gares SNCF françaises et les informations sur la régularité mensuelle des trains *Intercités* et *TGV*¹.

1 Prise en main

1. À partir du code SQL de création des tables et de spécification des contraintes, dessinez une modélisation relationnelle des éléments considérés.
2. Chargez la base de données “`sncf.db`”.

2 Interrogation de la base avec python

Nous allons maintenant déclencher des requêtes SQL en python.

Téléchargez le fichier `exemple.py` sur l'espace moodle et mettez-les dans le même répertoire.

Ce script vous montre comment on se connecte à la base de données `sqlite`, exécute une requête et récupère les résultats (deux exemples de requête : avec un résultat unique ou avec plusieurs lignes de résultats). Ce fichier pourra vous servir de base pour toutes les questions ci-dessous.

Remplissez la valeur de `SQLITE_DATABASE` en indiquant le chemin complet vers votre fichier `sncf.db`.

1. ★ Créez un programme python nommé `gares.py` qui se connecte à la base de données, récupère la liste des gares du département “Aude”, et l’affiche (avec la fonction `print`).

Le résultat sera donc :

```
> python gares.py
0087781104 : Narbonne
0087781088 : Leucate la Franqui
0087781062 : Port-la-Nouvelle
0087615286 : Carcassonne
...
```

(18 lignes)

2. ★★ Créez un script `coordonnees.py` créant et utilisant une fonction permettant d’obtenir les coordonnées (latitude et longitude) d’une gare dont le nom exact est passé en paramètre. La fonction sera donc définie ainsi :

```
## Recuperation des coordonnees d'une gare a partir
## d'une connexion a la base et du nom de la gare
def get_gare_info(database_cursor, gare_name):
```

1. Le code SQL fourni est une adaptation relationnelle des données fournies sur SNCF Open Data : <https://ressources.data.sncf.com/explore/>

```

# interrogation de la base pour obtenir
# la latitude et la longitude de la gare
...
return (latitude, longitude)

```

Ainsi :

```

> python coordonnees.py Annecy
(45.902, 6.12184)
> python coordonnees.py "Les Houches"
(45.8936, 6.79692)

```

3. ★★★ Le *taux de régularité* est le pourcentage de trains arrivés à l'heure sur une période donnée sur un trajet donné. Quels sont les 10 pires taux de régularité obtenus sur un mois (quel mois, quelle ligne, quel taux de régularité) ?
4. ★★★ Créez maintenant un script `retards_cumules.py` qui affiche toutes les informations sur les trains programmés, annulés, retardés à l'arrivée d'une gare passée en argument, ainsi que la somme totale et le taux de régularité.

Par exemple :

```

> python retards_cumules.py "Paris Gare de Lyon"
...
2017-05 Nice -> Paris Gare de Lyon : 213 programmés, 1 annulés, 51 retardés
2017-05 Saint-Étienne Châteaureux -> Paris Gare de Lyon : 106 programmés, 0 annulés, 10 retardés
2017-05 Nîmes -> Paris Gare de Lyon : 351 programmés, 2 annulés, 77 retardés
2017-05 Montpellier Saint-Roch -> Paris Gare de Lyon : 342 programmés, 2 annulés, 73 retardés
2017-05 Perpignan -> Paris Gare de Lyon : 147 programmés, 0 annulés, 48 retardés

Total trains programmés : 396749.0
Total trains annulés : 1519.0 (0.382861708536 %)
Total trains retardés : 45377.0 (11.4372058909 %)
Taux de régularité : 88.1799324006 %

```

5. ★★★★★ Créez un programme python `retards_region.py` fournissant la liste des retards de train au départ ou l'arrivée d'une gare d'une région passée en argument.

```

> python retards_region.py 'Occitanie'
2012-01 Perpignan -> Paris Gare de Lyon : 152 programmés, 0 annulés, 13 retardés
2012-01 Paris Gare de Lyon -> Nîmes : 357 programmés, 0 annulés, 39 retardés
2011-10 Paris Montparnasse -> Toulouse Matabiau : 118 programmés, 0 annulés, 45 retardés
2011-10 Montpellier Saint-Roch -> Paris Gare de Lyon : 394 programmés, 1 annulés, 44 retardés
2011-10 Paris Gare de Lyon -> Montpellier Saint-Roch : 380 programmés, 2 annulés, 36 retardés
2011-12 Montpellier Saint-Roch -> Paris Gare de Lyon : 336 programmés, 0 annulés, 30 retardés
2011-12 Paris Gare de Lyon -> Montpellier Saint-Roch : 323 programmés, 0 annulés, 24 retardés
2012-02 Montpellier Saint-Roch -> Lyon Part Dieu : 343 programmés, 1 annulés, 84 retardés
...

```

(1017 lignes)

3 Bonus 1 : Génération d'un fichier HTML

Nous allons maintenant déclencher des requêtes SQL en python et produire des résultats en HTML sur un petit serveur Web déployé pour l'occasion. Le CSS sera utilisé pour la mise en forme.

- Suivez pas à pas les instructions données ci-dessous pour lancer le serveur HTTP et vérifier qu'il fonctionne.
 - Placez le script `httpserver.py` et le fichier exemple `index.html` dans le même répertoire
 - Lancez le serveur : `python httpserver.py`
 - Vérifier le fonctionnement en allant sur localhost :8083 (le fichier `index.html` doit s'afficher). En pratique, localhost est la simulation de notre serveur, sur le port 8083 indiqué dans le script `httpserver.py`
 - `httpserver.py` déclare que le répertoire `cgi_bin` peut contenir des scripts python exécutables. Placez donc vos scripts python qui génèrent du HTML dans un répertoire `cgi_bin`
 - Rendez les scripts python exécutables (`chmod 777` ou par l'explorateur)
 - Un script python doit générer du HTML bien formé (affiché par exemple avec la commande `print`)
- Créez les programmes `gares.html.py` (★) et `retards_cumules.html.py` (★★★) reprenant les programmes `gares.html.py` et `retards_cumules.py`, et générant une page HTML présentant le résultat dans un tableau. La page HTML dynamiquement créée contiendra un lien vers une feuille de style CSS statique `style.css` contenant les instructions de mises en forme. Pour visualiser le résultat, on appellera donc dans notre navigateur les URL suivantes :

http://localhost:8083/cgi_bin/gares_html.py

et

http://localhost:8083/cgi_bin/retards_cumules_html.py?gare=Paris Gare de Lyon

Le résultat ressemblera à cela² :

Liste des gares de l'Aude	
Code UIC	Nom de la gare
0087598755	Limoux Flassian
0087615039	Castelnaudary
0087615054	Bram
0087615112	Lézignan-Corbières
0087615120	Couffoulens - Leuc
0087615138	Verzeille
0087615146	Pomas
0087615161	Limoux
0087615187	Alet-les-Bains
0087615203	Couiza - Montazels
0087615211	Espéaza
0087615229	Campagne-sur-Aude
0087615260	Quillan
0087615286	Carcassonne
0087781062	Port-la-Nouvelle
0087781088	Leucate la Franqui
0087781104	Narbonne
0087781161	Coursan

et

2. Les couleurs utilisées dans l'exemple sont #AF0034 pour le texte et #CCCCCC pour le fond.

Liste des retards à l'arrivée à la gare de Paris Gare de Lyon (taux de disponibilité : 88.18 %)					
Date	Départ	Arrivée	Programmés	Annulés	Retardés
2011-09	Bellegarde	Paris Gare de Lyon	257	0	13
2011-09	Avignon TGV	Paris Gare de Lyon	381	1	36
2011-09	Valence TGV Rhône-Alpes Sud	Paris Gare de Lyon	234	0	29
2011-09	Le Creusot - Montceau-les-Mines - Montchanin TGV	Paris Gare de Lyon	216	0	27
2011-09	Annecy	Paris Gare de Lyon	196	0	19
2011-09	Mâcon Loché TGV	Paris Gare de Lyon	180	0	16
2017-05	Lyon Part Dieu	Paris Gare de Lyon	628	3	68
2017-05	Marseille Saint-Charles	Paris Gare de Lyon	421	0	55
2017-05	Toulon	Paris Gare de Lyon	289	1	61
Total			396749	1519	45377

4 Bonus 2 : évolution de la régularité

Si vous avez plus de temps, créez en python des pages HTML montrant les statistiques disponibles sur une région, un axe ou une gare. Par exemple, une “fiche région” donnera la liste des gares de la région, avec les retards et l’évolution dans le temps des taux de régularité.

Pour la visualisation des statistiques, vous pourrez utiliser du texte, des tableaux ou, pour les plus motivés, des bibliothèques de visualisation de données comme [Google Charts](#) ou [amCharts](#) ou [matplotlib](#) pour les inconditionnels de python.

