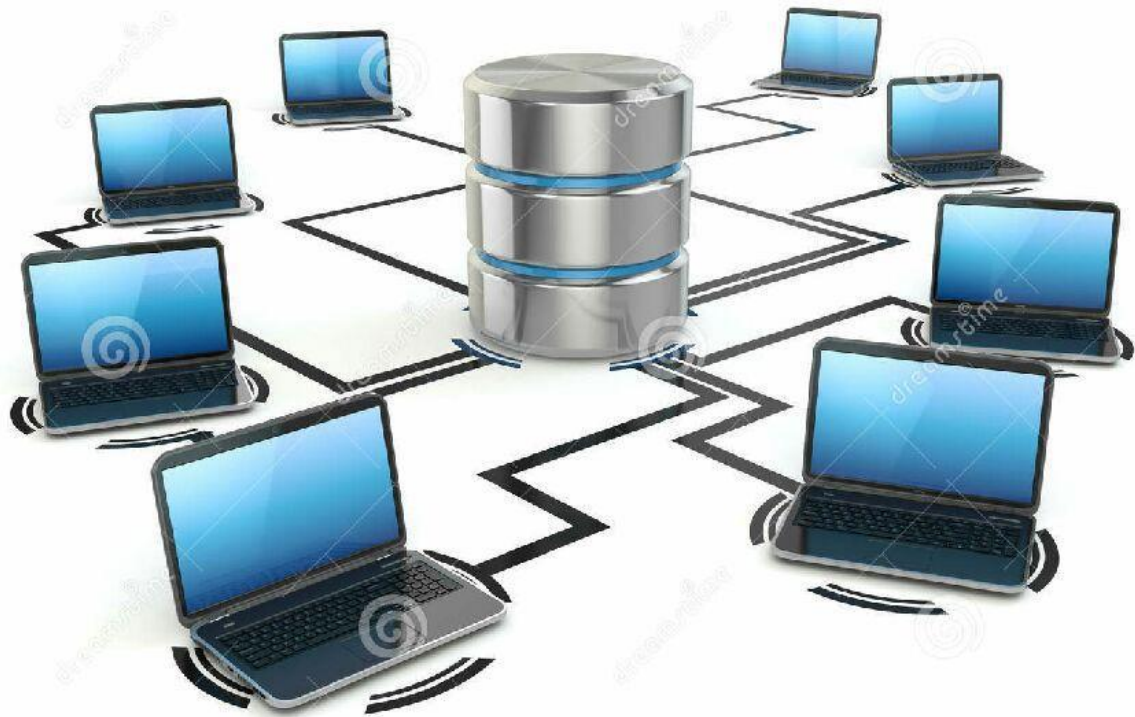


# Rapport de Projet



Michaëla Pierre- Louis & Arthy Uthayasuhanthan  
Groupe 107 et 109 | IUT Paris Descartes

## Table des matières

1.	Application.....	3
2.	Test.....	3
2.1.	Bilans de validation.....	3
2.2.	Les difficultés rencontrées.....	10
2.2.1.	La gestion du temps.....	10
2.2.2.	Le développement par sprint.....	10
2.2.3.	Les structures de données.....	10
2.2.4.	Les erreurs d'inattention.....	10
2.3.	Bilan de fin de projet.....	11
2.3.1.	Ce qui a été réussi.....	11
2.3.2.	Ce qui peut être amélioré.....	11
3.	Annexe.....	12
3.1.	Sprint 1 .....	13
3.2.	Sprint 2 .....	14
3.4	Sprint 3 .....	15
3.4	Sprint 4 .....	16
3.4	Sprint 5 .....	16

## 1. Application

L'application a pour but de programmer un interpréteur de commande pour la gestion d'une table de données numérique générique.

Les fonctionnalités du programme sont :

- Création d'une table de données et la définition de son schéma de donnée
- Destruction d'une table de données
- Affichage du schéma d'une table de données
- Insertion d'une ligne de données en fin de table
- Affichage d'un enregistrement d'un numéro donné
- Affichage de l'ensemble des lignes de données
- Destruction d'un enregistrement sélectionné par son numéro
- Sélection et affichage des lignes de données suivant un critère donné et d'un intervalle de sélection choisi
- Sorti de programme par la commande "Exit"

## 2. Tests

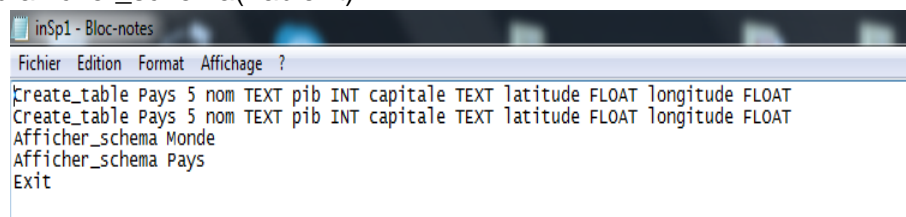
A travers notre projet nous avons dû effectuer des test selon une méthode particulière qui est celle de la méthode de développement agile. En effet après chaque nous avons eu l'obligation de tester la validité de nos sprint défini par des spécifications grâce à plusieurs test de redirection correspondant à des in et des out.

A travers le projet nous avons eu à faire face à 5 phase de test où chaque sprint devait permettre de coder et de tester plusieurs fonctions. La méthode de développement agile s'est donc révéler utile et efficace car elle nous permettait de se concentrer sur une partie précise de l'application et donc par la même occasion de réduire le nombre d'erreur pour la finition du projet .

### 2.1 Bilans de validation

**Objectif du sprint 1** : définir la structure de la table de données à travers le nom de la table , son nombre de champs , pour chaque champs définir un nom et son type. Et une fois tous les champs compléter de la table , affichage de la structure de la table de donnée. Il y avait deux fonctions à coder:

- void create\_table (Table \*t)
- void afficher\_schema(Table \*t)



```
inSp1 - Bloc-notes
Fichier Edition Format Affichage ?
Create_table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT
Create_table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT
Afficher_schema Monde
Afficher_schema Pays
Exit
```

Figure 1 : Jeu de données inSp1 du Sprint 1

Une fois notre sprint codé, nous avons pu le tester grâce à ce jeu de donné. Nous avons à notre disposition un résultat attendu sous la forme d'un fichier texte. Nous avons donc exécuté notre programme sous la console de commande de Windows avec comme entrée le fichier texte inSp1 (figure 1) et comme sortie notre fichier run1.txt (ci-dessous).

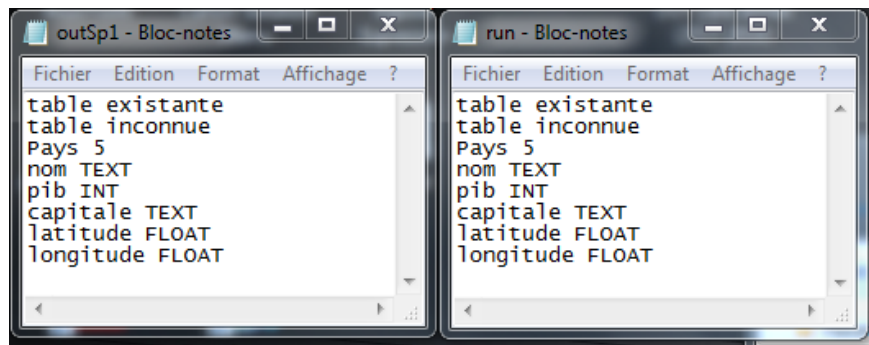


Figure 2 : Résultat attendu outSp1.txt (à gauche) et résultat run1.txt (à droite)

Comme nous le montre le résultat (figure 3), il n'y a aucune différence entre les deux fichiers, ce qui nous permet donc de conclure que notre sprint est 0-défaut par rapport à notre jeu de données (figure 1). Le sprint 1 validé, nous avons pu passer au second sprint.

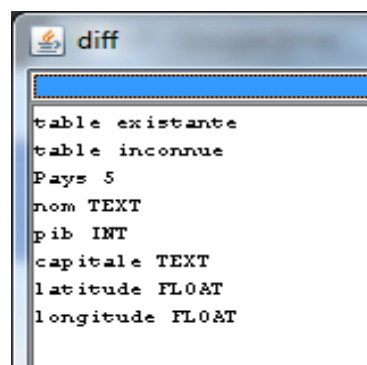


Figure 3 : résultat du diff entre run1.txt et outSp1.txt

**Objectif du sprint 2 :** Insérer une ligne de donnée en fin de table et afficher l'ensemble des lignes de données par ordre d'insertion. Il y avait deux fonctions à coder :

- void inserer\_enregistrement (Table \*t)
- void afficher\_enregistrement (Table\*t)

Le jeu de donné de test est légèrement le même que pour le sprint 1. Nous remplissons notre table avec des données réels de plusieurs pays accompagné de leurs différents caractéristiques notamment leur PIB , longitude , latitude, etc... (figure 4).



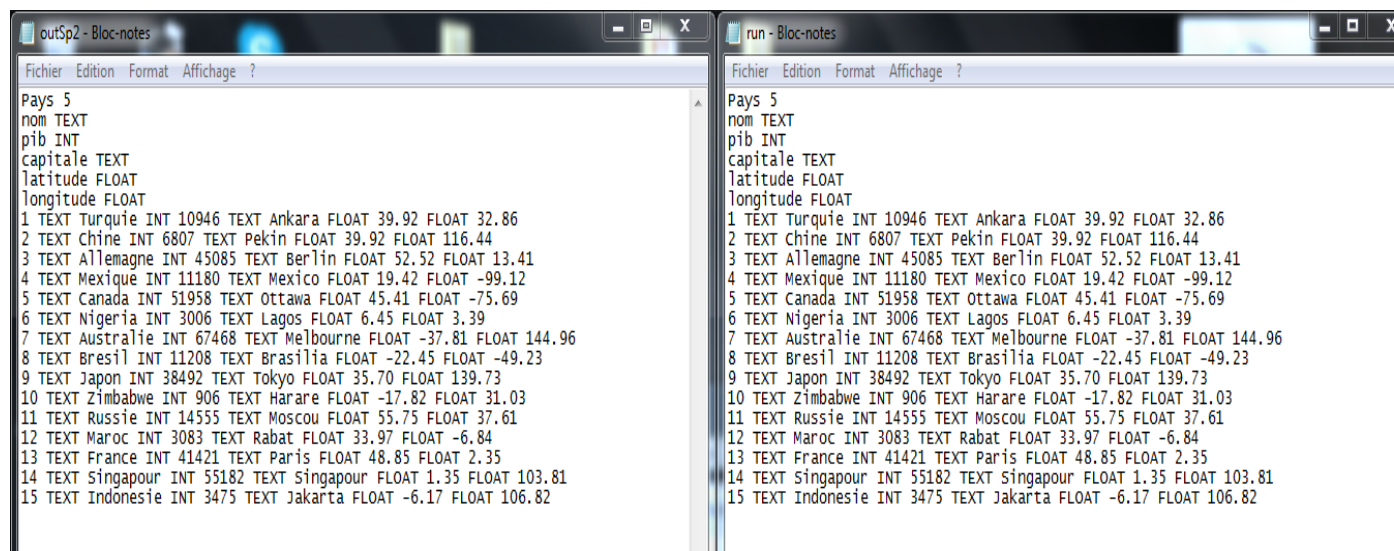
```

inSp2 - Bloc-notes
Fichier Edition Format Affichage ?
create_table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT
Afficher_schema Pays
Insert_enregistrement Pays Turquie 10946 Ankara 39.92 32.86
Insert_enregistrement Pays Chine 6807 Pekin 39.92 116.44
Insert_enregistrement Pays Allemagne 45085 Berlin 52.52 13.41
Insert_enregistrement Pays Mexique 11180 Mexico 19.42 -99.12
Insert_enregistrement Pays Canada 51958 Ottawa 45.41 -75.69
Insert_enregistrement Pays Nigeria 3006 Lagos 6.45 3.39
Insert_enregistrement Pays Australie 67468 Melbourne -37.81 144.96
Insert_enregistrement Pays Bresil 11208 Brasilia -22.45 -49.23
Insert_enregistrement Pays Japon 38492 Tokyo 35.70 139.73
Insert_enregistrement Pays Zimbabwe 906 Harare -17.82 31.03
Insert_enregistrement Pays Russie 14555 Moscou 55.75 37.61
Insert_enregistrement Pays Maroc 3083 Rabat 33.97 -6.84
Insert_enregistrement Pays France 41421 Paris 48.85 2.35
Insert_enregistrement Pays Singapour 55182 Singapour 1.35 103.81
Insert_enregistrement Pays Indonesie 3475 Jakarta -6.17 106.82
Afficher_enregistrements Pays
Exit

```

Figure 4 : Jeu de données inSp2 du sprint 2

Comme durant le sprint 1, nous avons testés notre programme sur ce jeu de donn  . Le r  sultat attendu se trouve dans la colonne de gauche tandis que le r  sultat de notre programme se trouve dans la colonne de droite.



```

outSp2 - Bloc-notes
Fichier Edition Format Affichage ?
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
15 TEXT Indonesie INT 3475 TEXT Jakarta FLOAT -6.17 FLOAT 106.82

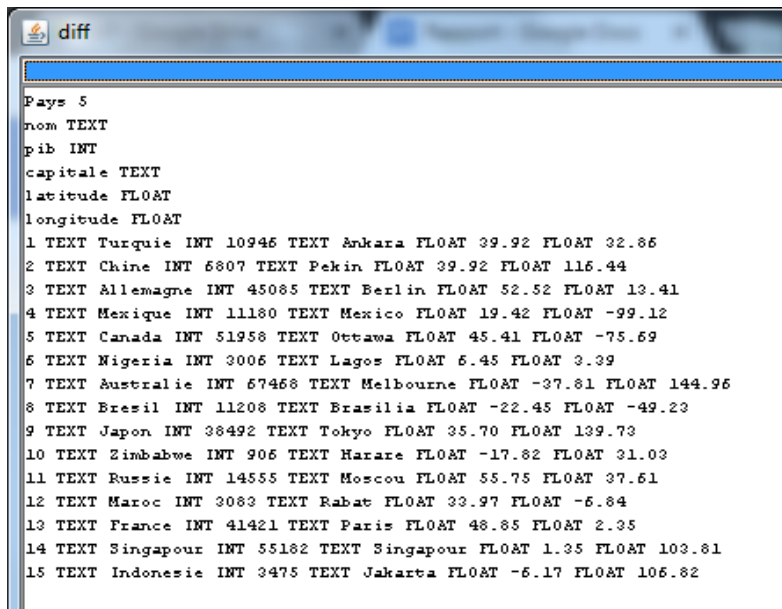
```

```

run - Bloc-notes
Fichier Edition Format Affichage ?
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
15 TEXT Indonesie INT 3475 TEXT Jakarta FLOAT -6.17 FLOAT 106.82

```

Figure 5 : R  sultat attendu outSp2.txt (   gauche) et r  sultat run2.txt (   droite)



```

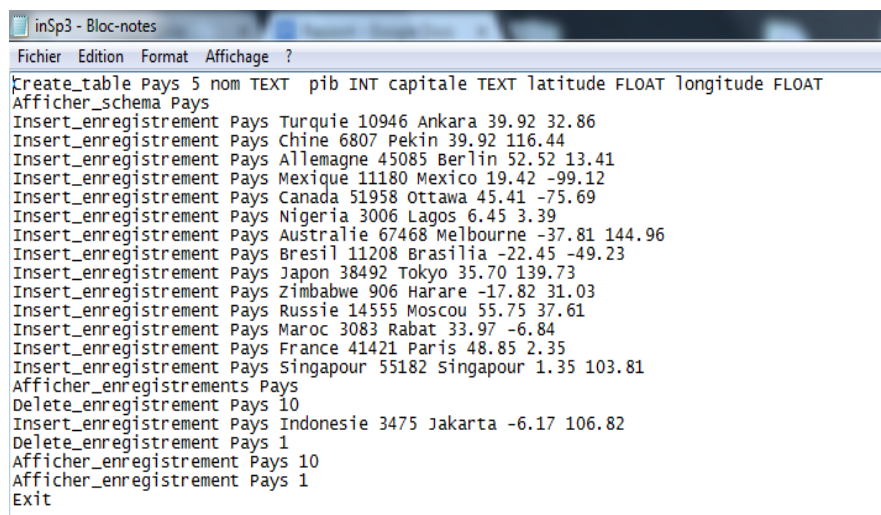
diff
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
15 TEXT Indonesie INT 3475 TEXT Jakarta FLOAT -6.17 FLOAT 106.82

```

Figure 6 : Résultat du diff entre run2.txt et outSp2.txt

**Objectif du Sprint 3 :** Suppression d'une ligne de données d'un numéro donné et affichage d'une ligne de donnée d'un numéro donné. Il y avait deux fonctions à coder :

- void delete\_enregistrement (Table \*t)
- void afficher\_enregistrement (Table \*t)



```

inSp3 - Bloc-notes
Fichier Edition Format Affichage ?
Create_table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT
Afficher_schema Pays
Insert_enregistrement Pays Turquie 10946 Ankara 39.92 32.86
Insert_enregistrement Pays Chine 6807 Pekin 39.92 116.44
Insert_enregistrement Pays Allemagne 45085 Berlin 52.52 13.41
Insert_enregistrement Pays Mexique 11180 Mexico 19.42 -99.12
Insert_enregistrement Pays Canada 51958 Ottawa 45.41 -75.69
Insert_enregistrement Pays Nigeria 3006 Lagos 6.45 3.39
Insert_enregistrement Pays Australie 67468 Melbourne -37.81 144.96
Insert_enregistrement Pays Bresil 11208 Brasilia -22.45 -49.23
Insert_enregistrement Pays Japon 38492 Tokyo 35.70 139.73
Insert_enregistrement Pays Zimbabwe 906 Harare -17.82 31.03
Insert_enregistrement Pays Russie 14555 Moscou 55.75 37.61
Insert_enregistrement Pays Maroc 3083 Rabat 33.97 -6.84
Insert_enregistrement Pays France 41421 Paris 48.85 2.35
Insert_enregistrement Pays Singapour 55182 Singapour 1.35 103.81
Afficher_enregistrements Pays
Delete_enregistrement Pays 10
Insert_enregistrement Pays Indonesie 3475 Jakarta -6.17 106.82
Delete_enregistrement Pays 1
Afficher_enregistrement Pays 10
Afficher_enregistrement Pays 1
Exit

```

Figure 5: Jeu de données inSp3 du sprint 3

Comme durant le sprint 2, nous avons testés notre programme sur ce jeu de donné. Le résultat attendu se trouve dans la colonne de gauche tandis que le résultat de notre programme se trouve dans la colonne de droite.

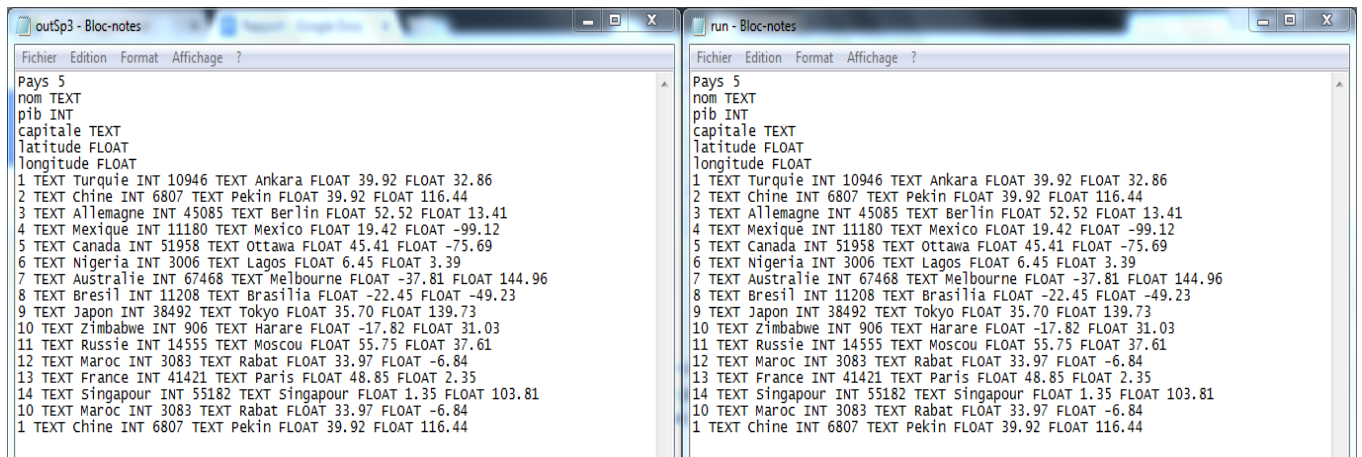


Figure 6: Résultat attendu outSp3.txt (à gauche) et résultat run3.txt (à droite)

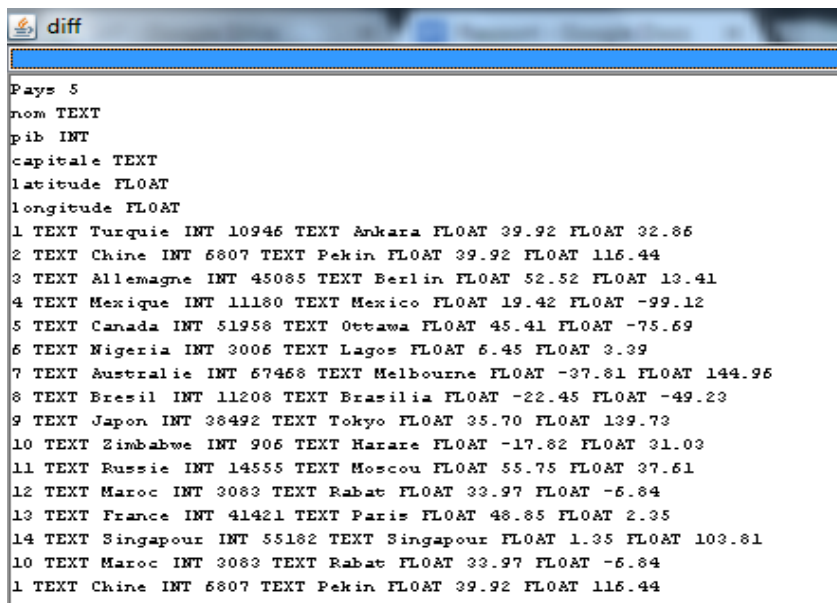


Figure 7 : Résultat du diff entre run3.txt et outSp3.txt

**Objectif du Sprint 4 :** Suppression d'une table de données d'un nom donné. Il y avait une fonction à coder :

- void delete\_table(Table \*t)



```

inSp4 - Bloc-notes
Fichier Edition Format Affichage ?
Create_Table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT
Afficher_schema Pays
Insert_enregistrement Pays Turquie 10946 Ankara 39.92 32.86
Insert_enregistrement Pays Chine 6807 Pekin 39.92 116.44
Insert_enregistrement Pays Allemagne 45085 Berlin 52.52 13.41
Insert_enregistrement Pays Mexique 11180 Mexico 19.42 -99.12
Insert_enregistrement Pays Canada 51958 Ottawa 45.41 -75.69
Insert_enregistrement Pays Nigeria 3006 Lagos 6.45 3.39
Insert_enregistrement Pays Australie 67468 Melbourne -37.81 144.96
Insert_enregistrement Pays Bresil 11208 Brasilia -22.45 -49.23
Insert_enregistrement Pays Japon 38492 Tokyo 35.70 139.73
Insert_enregistrement Pays Zimbabwe 906 Harare -17.82 31.03
Insert_enregistrement Pays Russie 14555 Moscou 55.75 37.61
Insert_enregistrement Pays Maroc 3083 Rabat 33.97 -6.84
Insert_enregistrement Pays France 41421 Paris 48.85 2.35
Insert_enregistrement Pays Singapour 55182 Singapour 1.35 103.81
Afficher_enregistrements Pays
Delete_enregistrement Pays 10
Insert_enregistrement Pays Indonesie 3475 Jakarta -6.17 106.82
Delete_enregistrement Pays 1
Afficher_enregistrements Pays 10
Delete_Table Pays
Create_Table Calendrier 3 evenement DATE description TEXT pays TEXT
Afficher_schema Calendrier
Insert_enregistrement Calendrier 14/10/1066 Bataille_Hastings Angleterre
Insert_enregistrement Calendrier 19/08/1274 Sacre_Edouard_I Angleterre
Insert_enregistrement Calendrier 07/10/1337 Guerre_Cent_Ans France
Insert_enregistrement Calendrier 07/06/1494 Traite_Tordesillas Espagne
Insert_enregistrement Calendrier 13/09/1515 Bataille_Marignan France
Insert_enregistrement Calendrier 10/08/1539 Ordonnance_Villiers_Cotterets France
Afficher_enregistrement Calendrier 2
Insert_enregistrement Calendrier 13/04/1598 Edit_Nantes France
Insert_enregistrement Calendrier 02/09/1666 Incendie_Londres Angleterre
Insert_enregistrement Calendrier 21/01/1793 Mort_Louix_XVI France
Insert_enregistrement Calendrier 28/05/1936 Machine_Turing Angleterre
Afficher_enregistrements Calendrier
Exit

```

Figure 8: Jeu de données inSp4 du sprint 4

Comme durant le sprint 3, nous avons testés notre programme sur ce jeu de donn  . Le r  sultat attendu se trouve dans la colonne de gauche tandis que le r  sultat de notre programme se trouve dans la colonne de droite.

```

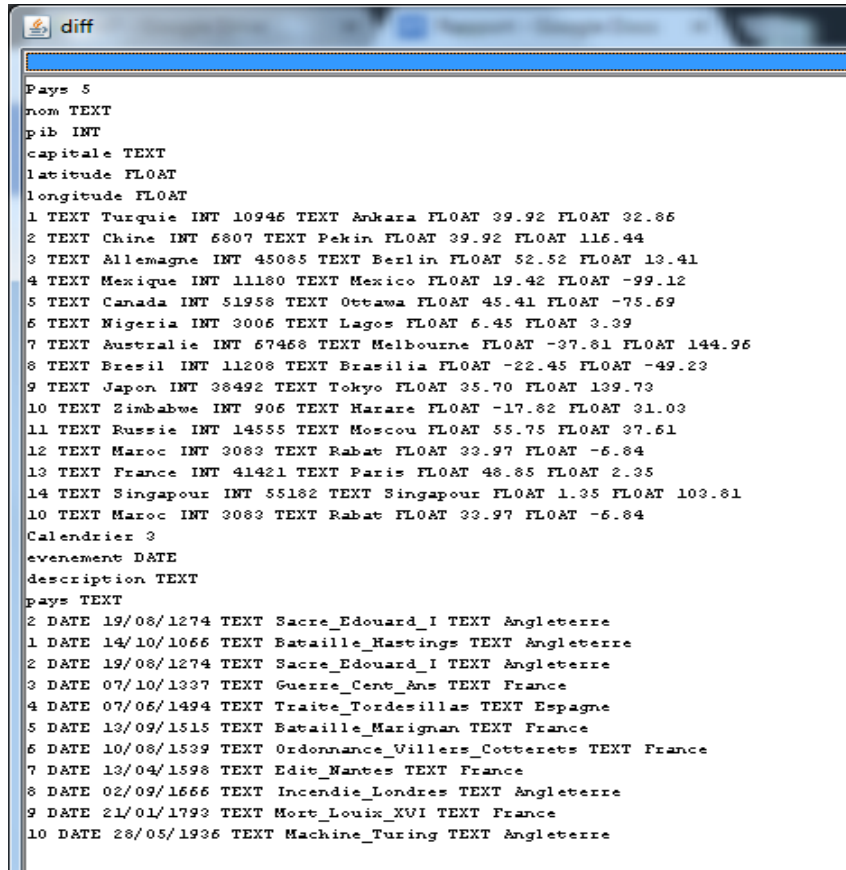
outSp4 - Bloc-notes
Fichier Edition Format Affichage ?
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
Calendrier 3
evenement DATE
description TEXT
pays TEXT
2 DATE 19/08/1274 TEXT Sacre_Edouard_I TEXT Angleterre
1 DATE 14/10/1066 TEXT Bataille_Hastings TEXT Angleterre
2 DATE 19/08/1274 TEXT Sacre_Edouard_I TEXT Angleterre
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_Villiers_Cotterets TEXT France
7 DATE 13/04/1598 TEXT Edit_Nantes TEXT France
8 DATE 02/09/1666 TEXT Incendie_Londres TEXT Angleterre
9 DATE 21/01/1793 TEXT Mort_Louix_XVI TEXT France
10 DATE 28/05/1936 TEXT Machine_Turing TEXT Angleterre

run - Bloc-notes
Fichier Edition Format Affichage ?
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
Calendrier 3
evenement DATE
description TEXT
pays TEXT
2 DATE 19/08/1274 TEXT Sacre_Edouard_I TEXT Angleterre
1 DATE 14/10/1066 TEXT Bataille_Hastings TEXT Angleterre
2 DATE 19/08/1274 TEXT Sacre_Edouard_I TEXT Angleterre
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_Villiers_Cotterets TEXT France
7 DATE 13/04/1598 TEXT Edit_Nantes TEXT France
8 DATE 02/09/1666 TEXT Incendie_Londres TEXT Angleterre
9 DATE 21/01/1793 TEXT Mort_Louix_XVI TEXT France
10 DATE 28/05/1936 TEXT Machine_Turing TEXT Angleterre

```

Figure 9: R  sultat attendu outSp4.txt (   gauche) et r  sultat run4.txt (   droite)





```

diff
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 5.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
Calendrier 3
evenement DATE
description TEXT
pays TEXT
2 DATE 19/08/1274 TEXT Sacre_Edouard_I TEXT Angleterre
1 DATE 14/10/1066 TEXT Bataille_Hastings TEXT Angleterre
2 DATE 19/08/1274 TEXT Sacre_Edouard_I TEXT Angleterre
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_Villers_Cotterets TEXT France
7 DATE 13/04/1598 TEXT Edit_Nantes TEXT France
8 DATE 02/09/1666 TEXT Incendie_Londres TEXT Angleterre
9 DATE 21/01/1793 TEXT Mort_Louis_XVI TEXT France
10 DATE 28/05/1936 TEXT Machine_Turing TEXT Angleterre

```

Figure 10 : Résultat du diff entre run4.txt et outSp4.txt

**Objectif du Sprint 5 :** Sélection et affichage des enregistrements suivant un champs donné et un intervalle de selection donné :

- void Select\_enregistrement(Table \*t)

```

inSp5 - Bloc-notes
Fichier Edition Format Affichage ?
Create_table Pays 5 nom TEXT pib INT capitale TEXT latitude FLOAT longitude FLOAT
Afficher_schema Pays
Insert_enregistrement Pays Turquie 10946 Ankara 39.92 32.86
Insert_enregistrement Pays Chine 6807 Pekin 39.92 116.44
Insert_enregistrement Pays Allemagne 45085 Berlin 52.52 13.41
Insert_enregistrement Pays Mexique 11180 Mexico 19.42 -99.12
Insert_enregistrement Pays Canada 51958 Ottawa 45.41 -75.69
Insert_enregistrement Pays Nigeria 3006 Lagos 6.45 3.39
Insert_enregistrement Pays Australie 67468 Melbourne -37.81 144.96
Insert_enregistrement Pays Bresil 11208 Brasilia -22.45 -49.23
Insert_enregistrement Pays Japon 38492 Tokyo 35.70 139.73
Insert_enregistrement Pays Zimbabwe 906 Harare -17.82 31.03
Insert_enregistrement Pays Russie 14555 Moscou 55.75 37.61
Insert_enregistrement Pays Maroc 3083 Rabat 33.97 -6.84
Insert_enregistrement Pays France 41421 Paris 48.85 2.35
Insert_enregistrement Pays Singapour 55182 Singapour 1.35 103.81
Afficher_enregistrements Pays
Delete_enregistrement Pays 10
Insert_enregistrement Pays Indonesie 3475 Jakarta -6.17 106.82
Delete_enregistrement Pays 1
Afficher_enregistrement Pays 10
Select_enregistrement Pays longitude -45 45
Delete_table Pays
Create_table Calendrier 3 evenement DATE description TEXT pays TEXT
Afficher_schema Calendrier
Insert_enregistrement Calendrier 14/10/1066 Bataille_Hastings Angleterre
Insert_enregistrement Calendrier 19/08/1274 Sacre_Edouard_I Angleterre
Insert_enregistrement Calendrier 07/10/1337 Guerre_Cent_Ans France
Insert_enregistrement Calendrier 07/06/1494 Traite_Tordesillas Espagne
Insert_enregistrement Calendrier 13/09/1515 Bataille_Marignan France
Insert_enregistrement Calendrier 10/08/1539 Ordonnance_villiers_Cotterets France
Insert_enregistrement Calendrier 13/04/1598 Edit_Nantes France
Insert_enregistrement Calendrier 02/09/1666 Incendie_Londres Angleterre
Insert_enregistrement Calendrier 21/01/1793 Mort_Louix_XVI France
Insert_enregistrement Calendrier 28/05/1936 Machine_Turing Angleterre
Select_enregistrement Calendrier pays D G
Select_enregistrement Calendrier evenement 07/10/1337 10/08/1539
Exit

```

Figure 11: Jeu de données inSp5 du sprint 5

Comme durant le sprint 4, nous avons testés notre programme sur ce jeu de donné. Le résultat attendu se trouve dans la colonne de gauche tandis que le résultat de notre programme se trouve dans la colonne de droite.

outSp5 - Bloc-notes

```

Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
2 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
5 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
9 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
11 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
Calendrier 3
evenement DATE
description TEXT
pays TEXT
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_villiers_Cotterets TEXT France
7 DATE 13/04/1598 TEXT Edit_Nantes TEXT France
9 DATE 21/01/1793 TEXT Mort_Louix_XVI TEXT France
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_villiers_Cotterets TEXT France

```

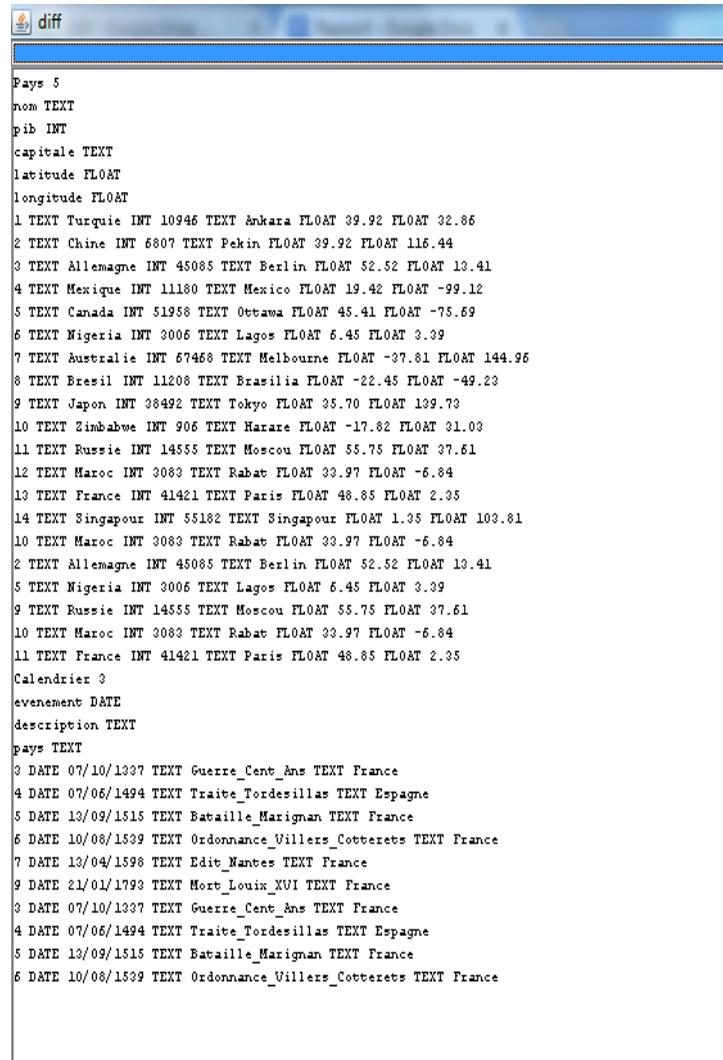
run - Bloc-notes

```

Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
2 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
5 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
9 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
11 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
Calendrier 3
evenement DATE
description TEXT
pays TEXT
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_villiers_Cotterets TEXT France
7 DATE 13/04/1598 TEXT Edit_Nantes TEXT France
9 DATE 21/01/1793 TEXT Mort_Louix_XVI TEXT France
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_villiers_Cotterets TEXT France

```

Figure 12: Résultat attendu outSp5.txt (à gauche) et résultat run5.txt (à droite)



```

diff
Pays 5
nom TEXT
pib INT
capitale TEXT
latitude FLOAT
longitude FLOAT
1 TEXT Turquie INT 10946 TEXT Ankara FLOAT 39.92 FLOAT 32.86
2 TEXT Chine INT 6807 TEXT Pekin FLOAT 39.92 FLOAT 116.44
3 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
4 TEXT Mexique INT 11180 TEXT Mexico FLOAT 19.42 FLOAT -99.12
5 TEXT Canada INT 51958 TEXT Ottawa FLOAT 45.41 FLOAT -75.69
6 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
7 TEXT Australie INT 67468 TEXT Melbourne FLOAT -37.81 FLOAT 144.96
8 TEXT Bresil INT 11208 TEXT Brasilia FLOAT -22.45 FLOAT -49.23
9 TEXT Japon INT 38492 TEXT Tokyo FLOAT 35.70 FLOAT 139.73
10 TEXT Zimbabwe INT 906 TEXT Harare FLOAT -17.82 FLOAT 31.03
11 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
12 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
13 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
14 TEXT Singapour INT 55182 TEXT Singapour FLOAT 1.35 FLOAT 103.81
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
2 TEXT Allemagne INT 45085 TEXT Berlin FLOAT 52.52 FLOAT 13.41
5 TEXT Nigeria INT 3006 TEXT Lagos FLOAT 6.45 FLOAT 3.39
9 TEXT Russie INT 14555 TEXT Moscou FLOAT 55.75 FLOAT 37.61
10 TEXT Maroc INT 3083 TEXT Rabat FLOAT 33.97 FLOAT -6.84
11 TEXT France INT 41421 TEXT Paris FLOAT 48.85 FLOAT 2.35
Calendrier 3
evenement DATE
description TEXT
pays TEXT
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_Villiers_Cotterets TEXT France
7 DATE 13/04/1598 TEXT Edit_Nantes TEXT France
9 DATE 21/01/1793 TEXT Mort_Louis_XVI TEXT France
3 DATE 07/10/1337 TEXT Guerre_Cent_Ans TEXT France
4 DATE 07/06/1494 TEXT Traite_Tordesillas TEXT Espagne
5 DATE 13/09/1515 TEXT Bataille_Marignan TEXT France
6 DATE 10/08/1539 TEXT Ordonnance_Villiers_Cotterets TEXT France

```

Figure 13 : Résultat du diff entre run5.txt et outSp5.txt

## 2.2. Les difficultés rencontrées

Nous avons jugé le démarrage comme étant plutôt difficile étant donnée que nous étions en pleine découverte des structures ce qui a conduit à la rencontre de certains problèmes lors de la phase de codage du premier sprint. En revanche le deuxième sprint a été légèrement moins contraignant que le sprint 1. Or, le sprint 2 n'était effectivement pas le plus contraignant car nous avons rencontrés des problèmes majeurs par rapport au codage du sprint 3 et du sprint 4.

### 2.2.1. La gestion du temps

Le temps a été une véritable contrainte pour nous, mais nous avons su gérer notre temps grâce à une bonne répartition du travail et une bonne communication entre nous.

### 2.2.2 Le développement par sprint

Le premier sprint nous a pris beaucoup de temps mais chaque sprint a coûté du temps or pas autant que le premier sprint. De plus, plus on avançait plus il fallait faire des retours en arrière sur les autres sprints car il fallait en effet revoir le codage des sprints précédents.

### 2.2.3. Les structures de données

Les structures de données se sont révélées être compliquées notamment à cause de leur organisation. En effet il nous a fallu beaucoup de temps pour comprendre comment les données étaient agencées surtout lorsque des structures étaient imbriquées dans d'autres structures comme par exemple la table "Champ" qui se trouve dans la table "Table".

### 2.2.4. Les erreurs d'inattention

Il nous est parfois arrivé de faire des erreurs d'inattention qui nous bloquaient lors de notre avancement. On peut notamment citer les fautes d'inattention qui ne sont pas détectées lors de la compilation du code (ex: "=" à la place de "==").

## 2.3. Bilan de fin de projet

### 2.3.1. Ce qui a été réussi

Les cinq sprints ont été réussi avec toutes les fonctionnalités demandées.

### 2.3.2. Ce qui peut être amélioré

En ce qui concerne le codage des fonctions Dans le but de gagner une certaine forme de fluidité et de clarté , nous avons eu l'idée de créer un fichier. Ce fichier se nommerait fonction.h et serait composé de toutes mes fonctions développé au cours du projet. Ce qui permettrait de créer une certaine agréabilité pour l'oeil et à lire puisqu'en effet il ne resterait plus que le main dans notre application.

De plus , nous avons pensés à réduire le temps d'exécution de notre application en procédant à l'imbrication des conditions puisque comme nous avons pu le voir , nous avons dû utiliser beaucoup de "if".

Enfin , par rapport au commande du programme nous avons pensés à la création de printf qui auront pour but de guider l'utilisateur. En effet notre programme actuel ne possède aucune indications ce qui est assez déstabilisant pour une personne qui ne connaît pas le but de notre programme.

### 3. Annexes

```
/* source.c
 * Auteurs : Arthy Uthayasuhanthan (109) & Michaëla Pierre-Louis (107)
 * Date de création : 02/10/17
 */
// Inclusions des en-têtes de bibliothèque
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <assert.h>
#include <string.h>
#include <locale.h>

#define _CRT_SECURE_NO_WARNINGS
#pragma warning (disable:4996)

// Déclarations des constantes
#define max_champs 25 // Nombre maximum de champs d'une table
#define lgMot 30 // Taille maximale d'une donnée
#define lgMax 80 // Taille maximale d'une ligne de commande
#define max_enregistrement 100 //Nombre maximum d'enregistrement d'une table

// Définitions des types structurés
// Informations relatives à la composition d'un champ
typedef struct {
    char nom[lgMot + 1]; // Nom du champ
    char type[lgMot + 1]; // Type du champ
} Champ;

// Structure d'une donnée
typedef struct {
    char data[lgMot]; // pour stocker les données qu'on a inséré
} Data;

/* la structure de donnée Enregistrement */
typedef struct {
    Data Pays[max_champs]; // Tableau de donné pour stocker les pays qu'on a inséré
} Enregistrement;

// Compostion d'une table
typedef struct {
    char nom[lgMot + 1]; // Nom de la table
    Champ schema[max_champs]; // Tableau de champ
```



```

        unsigned char nbChamps; // Nombre de champ créé
        Enregistrement PaysEnr[max_enregistrement];
        unsigned int indexPaysEnr; // Index des pays enregistrement
        unsigned int nbPays; // Nombre de pays enregistré a un instant donné inséré ajouté
        avec insert_enregistrement
    } Table;

/* la structure booléen */
typedef enum {
    false, true
} bool;

// Déclarations de fonction (prototypes)
/* Sprint 1 */
void create_table(Table* t);
void afficher_schema(Table* t);
/* Sprint 2 */
void insert_enregistrement(Table* t);
void afficher_enregistrements(Table* t);
/* Sprint 3 */
void delete_enregistrement(Table* t);
void afficher_enregistrement(Table* t); //
/* Sprint 4 */
void delete_table(Table* t);
/* Sprint 5 */
void Select_enregistrement(Table* t);

// Fonctions :
/*-----SPRINT 1-----*/
// Définitions de fonction
/* Définition du nom de la table , du nombre de champs , pour chaque champ son nom et
son type
* t [in-out] nom de la table , du nombre de champs , pour chaque champ son nom et son
type

*/
void create_table(Table* t) {
    char mot[lgMot + 1]; // sert à stocker les données que l'on va entrer dans la fenêtre de
la commande
    scanf("%s", &mot); // lecture des données que l'on a stocké dans la variable "mot"
    // Condition pour une table de donné existante
    if (strcmp(t->nom, mot) == 0) {
        printf("table existante\n");
    }
    // Sinon
    else {

```

```

        strcpy(t->nom, mot); // Copie du nom de la table dans la variable "mot"
scanf("%s", &mot);
t->nbChamps = atoi(mot); // convertis le nbChamps en int avec la fonction atoi

// Une boucle pour accéder aux noms et aux types de la structure Champ
qu'on a crée
for (unsigned int i = 0; i < t->nbChamps; ++i) {
    scanf("%s", &mot);
    strcpy(t->schema[i].nom, mot); // sert à copier les noms du Champ
dans le "mot"
    scanf("%s", &mot);
    strcpy(t->schema[i].type, mot); // sert à copier les types du Champ
dans le "mot"
}
}
}
/* Afficher structure d'une table de donnée
 * t [in] Nom et Type de chaque champ
 */
void afficher_schema(Table* t) {
    char mot[lgMot + 1]; // sert à stocker les données que je vais entrer dans la fenêtre de
la commande
    scanf("%s", &mot); // lecture des données stockées dans la variable "mot"
    // Condition pour une table inconnue
    if (strcmp(t->nom, mot) != 0) {
        printf("table inconnue\n");
    }
    // Sinon
    else {
        printf("%s %u\n", t->nom, t->nbChamps); // afficher le nom de la table et le
nombre de champs de la table
        // Une boucle pour afficher les noms et les types de champs
        for (unsigned int i = 0; i < t->nbChamps; i++) {
            printf("%s %s\n", t->schema[i].nom, t->schema[i].type);
        }
    }
}

/*-----SPRINT 2-----*/

/* Insérer une ligne de donnée en fin de table
 * t [in-out] Nom et Type de chaque donnée inséré dans la table
 */
void insert_enregistrement(Table* t) {
    char mot[lgMot + 1]; // sert à stocker les données qu'on va entrer dans la fenêtre de la
commande
    scanf("%s", &mot);
    if (strcmp(mot, t->nom) != 0) {

```

```

        printf("table inconnue");
    }

    else {
        for (unsigned int i = 0; i < t->nbChamps; ++i) {
            scanf("%s", &mot);
            strcpy(t->PaysEnr[t->nbPays].Pays[i].data, mot);
        }
        t->nbPays++;
        t->indexPaysEnr++;
    }
}

/* Afficher une ligne donné d'un numéro demandé d'un tableau de donnée
 * t [in] Ligne donné d'un numéro demandé
 */
void afficher_enregistrements(Table* t) {
    char mot[lgMot + 1];
    scanf("%s", &mot);

    if (strcmp(mot, t->nom) != 0) {
        printf("table inconnue");
    }

    else {
        for (unsigned int k = 1; k < t->indexPaysEnr;) {
            for (unsigned int j = 0; j < t->nbPays; ++j) {
                printf("%u", k);
                for (unsigned int i = 0; i < t->nbChamps; ++i) {
                    printf(" %s %s", t->schema[i].type, t-
>PaysEnr[j].Pays[i].data);
                }
                printf("\n");
                ++k;
            }
        }
    }
}

/*-----SPRINT 3-----*/
/*Suppression d'une donnée d'un numéro donné
 * t [in -out] Suppression d'une ligne d'une table de donné
 */
void delete_enregistrement(Table* t) {
    //condition pour les rangs des inserts
    //Modification de l'index
    //condition sur la taille de tableau après avoir supprimé la donnée
    char mot[lgMot + 1];

```

```

scanf("%s", &mot);

if (strcmp(mot, t->nom) != 0) {
    printf("table inconnue");
}
else {
    char mot[lgMot + 1];
    scanf("%s", &mot);
    unsigned int nb = atoi(mot);

    if (t->PaysEnr[nb].Pays == NULL) {
        printf("Enregistrement inconnu\n");
        return;
    }
    for (unsigned int j = nb; j <= t->nbPays; j++) {
        for (unsigned int i = 0; i < t->nbChamps; ++i) {
            strcpy(t->PaysEnr[j - 1].Pays[i].data, t-
>PaysEnr[j].Pays[i].data);
        }
        ++j;
    }
}
t->nbPays--;
t->indexPaysEnr--;
}

/* Afficher une ligne donné d'un numéro demandé d'un tableau de donnée
 * t [in] Ligne donné d'un numéro demandé
 */
void afficher_enregistrement(Table* t) {
    char mot[lgMot + 1];
    scanf("%s", &mot);

    if (strcmp(mot, t->nom) != 0) {
        printf("table inconnue");
    }

    else {
        scanf("%s", &mot);
        unsigned int nombre = atoi(mot);
        if (nombre > t->nbPays) {
            printf("Erreur le pays selectionne n'est pas enregistre\n");
        }

        else {
            printf("%u", nombre);
            for (unsigned int i = 0; i < t->nbChamps; ++i) {

```

```

                                printf(" %s %s", t->schema[i].type, t->PaysEnr[nombre -
1].Pays[i].data);
                                }
                                printf("\n");
                        }
                }
        }
}

```

```

/*-----SPRINT 4-----*/

```

```

/*Suppression d'une table de donné d'un nom donné

```

```

* t [in -out] Suppression d'une table

```

```

*/

```

```

void delete_table(Table* t) {
    char mot[lgMot + 1];
    scanf("%s", mot);

    if (strcmp(mot, t->nom) != 0) {
        printf("table inconnue\n");
    }

    else {

        for (unsigned int k = 0; k < t->nbChamps; ++k) {
            strcpy(t->schema[k].nom, "\0");
            strcpy(t->schema[k].type, "\0");
        }

        for (unsigned int j = 0; j < t->nbPays;) {
            for (unsigned int i = 0; i < t->nbChamps; ++i) {
                strcpy(t->PaysEnr[j].Pays[i].data, "\0");
            }
            ++j;
        }

    }

    t->indexPaysEnr = 0;
    t->nbPays = 0;
    t->nbChamps = 0;
    strcpy(t->nom, "\0");
    t = NULL;
}

```

```

/*-----SPRINT 5-----*/

```

```

/*Sélection d'une donnée dans un intervalle donnée

```

```

* t [in -out] Sélection d'une donnée

```

```

*/

```

```

void select_enregistrement(Table* t) {
    char mot[lgMot + 1];

```

```

char borne1[IgMot];
char borne2[IgMot];
char champ[IgMot];
char type[IgMot];
scanf("%s", mot);

if (strcmp(mot, t->nom) != 0) {
    printf("table inconnue\n");
}

scanf("%s", mot); //scan du champ de la donnee
strcpy(champ, mot);

for (unsigned int i = 0; i < t->nbChamps; ++i) {
    if (strcmp(t->schema[i].nom, champ) == 0) {
        strcpy(type, t->schema[i].type);
    }
}

scanf("%s", mot);
strcpy(borne1, mot);
scanf("%s", mot);
strcpy(borne2, mot);

for (unsigned int i = 0; i < t->nbPays; ++i) {
    for (unsigned int j = 0; j < t->nbChamps; ++j) {
        if (strcmp(champ, t->schema[j].nom) == 0) {
            if (compare_enregistrement(t->PaysEnr[i].Pays[j].data, type,
borne1, borne2) == true) {
                printf("%u", i + 1);
                for (unsigned int s = 0; s < t->nbChamps; ++s) {
                    printf(" %s %s", t->schema[s].type, t-
>PaysEnr[i].Pays[s].data);
                }
                printf("\n");
            }
        }
    }
}

}

/*
* data [in], le champ de donnée correspondant à la selection
* type [in], le type du champ de selection
* borne1 [in], la première borne de selection
* borne2 [in], la deuxième borne de selection

```



\* retour true si l'enregistrement est sélectionné, false sinon

\*/

```
bool compare_enregistrement(const char* data, const char* type, const char* borne1, const char* borne2) {
```

```
    if (strcmp(type, "INT") == 0) {
```

```
        return compare_enregistrement_int(data, borne1, borne2);
```

```
    }
```

```
    else if (strcmp(type, "DATE") == 0) {
```

```
        return compare_enregistrement_date(data, borne1, borne2);
```

```
    }
```

```
    else if (strcmp(type, "FLOAT") == 0) {
```

```
        return compare_enregistrement_float(data, borne1, borne2);
```

```
    }
```

```
    else {
```

```
        return compare_enregistrement_text(data, borne1, borne2);
```

```
    }
```

```
}
```

```
bool compare_enregistrement_int(const char* data, const char* borne1, const char* borne2)
```

```
{
```

```
    if (atoi(data) >= atoi(borne1) && atoi(data) <= atoi(borne2)) {
```

```
        return true;
```

```
    }
```

```
    else {
```

```
        return false;
```

```
    }
```

```
}
```

```
bool compare_enregistrement_float(const char* data, const char* borne1, const char*
```

```
borne2) {
```

```
    if (atof(data) >= atof(borne1) && atof(data) <= atof(borne2)) {
```

```
        return true;
```

```
    }
```

```
    else {
```

```
        return false;
```

```
    }
```

```
}
```

```
bool compare_enregistrement_text(const char* data, const char* borne1, const char*
```

```
borne2) {
```

```
    if ((strcmp(data, borne1) > 0) && (strcmp(data, borne2) < 0)) {
```

```
        return true;
```

```
    }
```

```

        else {
            return false;
        }
    }

bool compare_enregistrement_date(const char* data, const char* borne1, const char*
borne2) {
    if ((strncmp(data + 6, borne1 + 6, 4) >= 0) && (strncmp(data + 6, borne2 + 6, 4) <=
0)) {
        if ((strncmp(data + 3, borne1 + 3, 2) >= 0) || (strncmp(data + 3, borne2 + 3, 2)
<= 0)) {
            if ((strncmp(data, borne1, 2) >= 0) || (strncmp(data, borne2, 2) <= 0)) {
                return true;
            }
        }
    }

    else {
        return false;
    }
}

```

// Main

```

int main() {
    Table t;
    char mot[lgMax + 1];
    t.nbPays = 0;
    t.indexPaysEnr = 0;

    do { // Boucle infini sur 6 commandes
        scanf("%s", mot); // Mot lu considéré comme une chaîne de caractère
        if (strcmp(mot, "Create_table") == 0) {
            create_table(&t);
        }

        if (strcmp(mot, "Afficher_schema") == 0) {
            afficher_schema(&t);
        }

        if (strcmp(mot, "Insert_enregistrement") == 0) {
            insert_enregistrement(&t);
        }

        if (strcmp(mot, "Afficher_enregistrements") == 0) {
            afficher_enregistrements(&t);
        }
    }
}

```

```
        if (strcmp(mot, "Delete_enregistrement") == 0) {
            delete_enregistrement(&t);
        }

        if (strcmp(mot, "Afficher_enregistrement") == 0) {
            afficher_enregistrement(&t);
        }
        if (strcmp(mot, "Delete_table") == 0) {
            delete_table(&t);
        }
        if (strcmp(mot, "Select_enregistrement") == 0) {
            select_enregistrement(&t);
        }
        if (strcmp(mot, "Exit") == 0) {
            exit(0);
        }

    } while (1);
    system("pause");
    return 0;
}
```