



MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE



Ministère de l'Éducation Nationale et de la Jeunesse



Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE
DE L'ÉDUCATION
NATIONALE ET
DE LA JEUNESSE

RAPPORT DE STAGE

TUTRICE DE STAGE : **THU-DUNG NGUYEN**

RESPONSABLE PÉDAGOGIQUE : **CÉLINE TOUILLON**

AFFECTATION : **CENTRE DE L'INFORMATIQUE STATISTIQUE ET DE
L'AIDE À LA DÉCISION**
(DEPP CISAD)

Barème de notation du rapport de stage (à insérer en première page du rapport de stage)

Nom de l'étudiant

	Très insuffisant	Insuffisant	Assez bien	Très bien
Respect des consignes	0	0.5	1	2
Mise en page et Orthographe				
Mise en page	0	0	0.5	1
Orthographe	0	0	0	1
Présentation Structuration...	0	0	0.5	1
Evaluation du contenu				
Sommaire et remerciements	0	0	0.5	1
Présentation de l'entreprise	0	0.5	1	2
Présentation du thème du stage	0	0.5	1	2
Analyse de l'existant et des besoins	0	1	2	3
Recherche et mise en place des solutions	0	1	2	3
Conclusion	0	0.5	1	2
Rédaction	0	0.5	1	2

Rapport remis le		Pénalités de retard	
------------------	--	---------------------	--

Appréciation globale	Note
	/ 20

Professeur ayant suivi le stage :

Table des matières

Présentation du rapport de stage

Barème	2
Table des matières	3
Remerciements	4

Structuration du stage

Présentation de l'entreprise	5
Présentation de la mission	6
Analyse de l'existant et des besoins	7

Réalisation de la mission

Choix de l'environnement	8
Présentation des bases de données	9
Extrait de code et explications	10

Bilan du rapport de stage

Résultat de la mission	15
Enseignement du stage	15
Difficultés rencontrées	15
Conclusion	16



Remerciements

J'aimerais tout d'abord remercier **Olivier THIÉBAUT**, chef de projet DevOps National au sein du Ministère de l'Éducation Nationale et de la Jeunesse pour m'avoir fait confiance ainsi que **Thu-Dung NGUYEN** cheffe de projet informatique au sein du centre DEPP -- CISAD du Ministère de l'Éducation Nationale et de la Jeunesse pour avoir accepté de me prendre sous son aile durant cette période de cinq semaines.

Merci à l'UGARH d'avoir accepté mon dossier et je remercie toutes les personnes avec lesquelles j'ai pu échanger et travailler, je pense notamment au CISAD, et ses équipes PRODUCTION et SYSTÈME.

Je remercie également mes professeurs du lycée Parc des Loges, Mme **ROLLAND** et Mme **TOUILLON** qui m'ont apporté les connaissances nécessaires tout au long de l'année pour me permettre de mener à bien ce stage.



Présentation de l'entreprise

Le ministère de l'Éducation Nationale et de la Jeunesse est l'administration française chargée de préparer et mettre en œuvre la politique du gouvernement dans les domaines de l'éducation. Il est dirigé par le ministre **Jean-Michel Blanquer**.

La direction de l'évaluation, de la prospective et de la performance (**DEPP**) exerce une fonction de suivi statistique, d'expertise et d'assistance pour le ministère de l'Éducation Nationale ainsi que pour le Ministère de l'Enseignement Supérieur et de la Recherche et de l'innovation. Elle garantit la qualité de la production statistique.

Le **CISAD** (centre de l'informatique statistique et de l'aide à la décision) assure une mission d'appui à l'information décisionnelle et statistique de la direction. Il assiste les maîtrises d'ouvrage dans le cadre d'équipes-projet. Il assure un rôle d'interface et de conseil dans les relations avec les autres structures de maîtrise d'œuvre de la centrale et des services déconcentrés.

La direction présente actuellement un effectif d'environ 150 employés, dans tous les secteurs compris et dont les locaux sont situés au : 61-65 rue Dutot - 75015 Paris.

Présentation de la mission

Dans le cadre de mon cursus, j'ai été amené à réaliser un stage de cinq semaines en entreprise dans le domaine informatique, le Ministère de l'Éducation Nationale m'a donc ouvert ses portes.

La mission principale du stage est **la migration d'une base de données Oracle vers une base de données PostgreSQL**. Cette base est utilisée par une application web¹ sur le suivi des élèves en dispositifs relais².

Le ministère a pour but actuellement de migrer l'ensemble de ses bases sous Oracle (licence payante) vers un produit libre (PostgreSQL).

La mission de migration a été découpée en trois parties :

1. **Migrer** premièrement toutes les tables de la base de données.
Proposition : Migrer les structures (les tables et les vues*)
2. **Migrer** ensuite toutes les vues³ de la base de données.
Proposition : Migrer les données
3. **Adapter** le code applicatif (en langage PHP) pour un bon fonctionnement de l'application web.

¹ : (Une application web est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes.)

² : (Les dispositifs relais sont des classes et des ateliers mis en place pour accueillir temporairement les élèves en rejet scolaire, au collège ou éventuellement au lycée, afin de leur redonner goût à l'école puis de les réintégrer dans le système scolaire ordinaire.)

³ : Une vue est une table virtuelle, c'est-à-dire dont les données ne sont pas stockées dans une table de la base de données, et dans laquelle il est possible de rassembler des informations provenant de plusieurs tables.

Pour effectuer ces tâches, je suis évidemment parti de l'existant, la base de données Oracle et le code applicatif en PHP qui contenaient

- Le code source de l'application web;
- Toutes les fonctions applicatives php;
- Toutes les requêtes SQL déjà préparées dans des fonctions.

Analyse de l'existant et des besoins

Lors de l'analyse des besoins, il a été identifié qu'il fallait créer la base de données PostgreSQL et qu'il fallait y intégrer les données contenues dans les tables et les vues de celle de la base de données ORACLE.

J'ai créé les tables de la base de données sous PostgreSQL et récupéré les données sans difficulté. En revanche, la migration des vues a été plus complexe. C'était un nouveau terrain inconnu pour moi, j'ai dû avant toute chose me documenter apprendre, et savoir ce que je manipule avant toute migration.

Le code PHP de l'application est basé sur le modèle d'architecture MVC (Modèle-Vue-Contrôleur) qui permet d'organiser son code source en trois parties :

- Données (modèle);
- Traitement (contrôleur);
- Présentation (vue).

L'avantage premier du MVC est la séparation des tâches, du métier et de l'interface utilisateur qui permet une visibilité nettement meilleure qu'un dossier qui regroupe la totalité des fichiers dedans.

Lors de la remise du dossier source de l'application web, je n'étais pas mis dans une position indélicate, le code applicatif intégral de l'application et de tous les fichiers ont très bien été documentés et commentés pour la bonne compréhension des développeurs qui travailleront dessus plus tard.

Choix de l'environnement

J'ai eu mes repères directement en prenant connaissance de chaque document et j'ai pu commencer à travailler sans trop de problèmes. Pour mener à bien la mission de mon stage, on m'a fait découvrir plusieurs logiciels (logiciels) pour la gestion et la migration des bases de données.



Le principe de base d'**ora2pg** est la découverte ou le scan automatique d'une base Oracle pour en extraire la structure ou les données et les convertir dans la syntaxe PostgreSQL



DBeaver est un client SQL multi-plateformes en Java II qui supporte les moteurs de bases de données les plus populaires et permet la connexion en simultanée de plusieurs bases de données à la fois.

Pour ne pas précipiter sans comprendre le travail demandé, savoir ce que je manipule et ce que je migre, j'ai donc utilisé le logiciel **DBeaver** qui permet une simplicité aux développeurs pour la migration. Grâce à la connexion en simultanée de plusieurs bases de données à la fois, alors que **ora2pg** permet une migration instantanée de toutes les tables et vues en accordant la bonne syntaxe.

J'ai donc pu me connecter à la base de données **ORACLE** et celle de **PostgreSQL** pour prendre connaissance des tables et des vues.

Présentation des bases de données

- ▷  ACADEMIE
- ▷  ACADEMIE_DEPARTEMENT
- ▷  CREL_ADMIN
- ▷  CREL_CHAMP
- ▷  CREL_CHAMP_TMP
- ▷  CREL_CORRESP_ACAD
- ▷  CREL_CORRESP_PJJ
- ▷  CREL_CORRESP_PJJ_OLD
- ▷  CREL_DATE_SUIVI
- ▷  CREL_MODULE
- ▷  CREL_NOMENCLATURE
- ▷  CREL_NOMENC_CLASSE
- ▷  CREL_ORIENTATION
- ▷  CREL_PERSONNEL
- ▷  CREL_PERSONNEL_ELEVE
- ▷  CREL_PERSONNEL_PP
- ▷  CREL_PERSONNEL_QS
- ▷  CREL_PJJ_DEPARTEMENT
- ▷  CREL_RELANCE
- ▷  CREL_SAISIE
- ▷  CREL_SAISIE_2011
- ▷  CREL_SAISIE_2011_SAV
- ▷  CREL_SAISIE_TMP
- ▷  LISTE_DISPO_RELAIS

Les tables principales depuis la base de données ORACLE qui sont à migrer sur la base de données PostgreSQL, sans compter les tables d'archives des années précédentes. (soit 24 tables*le nombre d'années archivées).

Les tables principales peuvent comporter jusqu'à ~82 champs dans la table en elle-même mais aucunes n'avaient de clés étrangères, seulement quelques unes avaient des clés primaires, et pratiquement toutes avaient une contrainte pour vérifier que certaines valeurs ne sont pas nulles.

La partie la plus importante du travail au niveau de la migration des bases de données est l'importation des vues d'une base à l'autre.

Il y avait plus ou moins 172 vues (en ne comptant pas les vues archivées des dernières années) à migrer sur la base de données PostgreSQL, tout en adaptant le code applicatif pour qu'il soit interprété et compilé avec la base de données PostgreSQL.

▷ V_1718_RETARD_ATE	▷ V_1819_CONSEIL_DISC_APRES_ATE
▷ V_1718_RETARD_CLA	▷ V_1819_CONSEIL_DISC_APRES_CLA
▷ V_1718_RETARD_INT	▷ V_1819_CONSEIL_DISC_APRES_INT
▷ V_1718_SITU1AN	▷ V_1819_CONSEIL_DISC_AVANT_ATE
▷ V_1718_SITU6MOIS	▷ V_1819_CONSEIL_DISC_AVANT_CLA
▷ V_1718_SITUFINAN	▷ V_1819_CONSEIL_DISC_AVANT_INT
▷ V_1718_SITUSORTIE	▷ V_1819_DERNIERE_CLASSE_ATE
▷ V_1819_AGE_ATE	▷ V_1819_DERNIERE_CLASSE_CLA
▷ V_1819_AGE_CLA	▷ V_1819_DERNIERE_CLASSE_INT
▷ V_1819_AGE_INT	▷ V_1819_DR_AVEC_REPONSE
▷ V_1819_AGE_SEJOUR_ATE	▷ V_1819_DUREE_ALTERNANCE_ATE
▷ V_1819_AGE_SEJOUR_CLA	▷ V_1819_DUREE_ALTERNANCE_CLA
▷ V_1819_AGE_SEJOUR_INT	▷ V_1819_DUREE_ALTERNANCE_INT
▷ V_1819_AGE_TMP_ATE	▷ V_1819_DUREE_ATE
▷ V_1819_AGE_TMP_CLA	▷ V_1819_DUREE_CLA
▷ V_1819_AGE_TMP_INT	▷ V_1819_DUREE_DESCOL_ATE
▷ V_1819_AGE_TMP_SEJOUR_ATE	▷ V_1819_DUREE_DESCOL_CLA
▷ V_1819_AGE_TMP_SEJOUR_CLA	▷ V_1819_DUREE_DESCOL_INT
▷ V_1819_AGE_TMP_SEJOUR_INT	▷ V_1819_DUREE_DESCOL_TMP_ATE
	▷ V_1819_DUREE_DESCOL_TMP_CLA
	▷ V_1819_DUREE_DESCOL_TMP_INT
	▷ V_1819_DUREE_INT

Une partie de la liste des vues à migrer sont celles-ci, la liste n'est pas complète.
(à titre informatif)

Extrait de code et explications

Lors de la migration de la base de données et plus essentiellement des vues, d'une base à une autre, les fonctions ne s'écrivent pas de la même façon, il fallait donc adapter le code SQL en question pour qu'il fonctionne sous la nouvelle base de données.

→ Exemple pour illustrer :

```
SELECT academie,
       departement,
       uai,
       age,
       age AS tranche,
       DECODE (SIGN (REPLACE (derniere_classe, 24, NULL) - 5),
              1, 6,
              REPLACE (derniere_classe, 24, NULL))
       AS derniere_classe,
       retard
FROM v_1718_retard_tmp_int
WHERE age BETWEEN 12 AND 15 OR age IS NULL
```

```
select
  "academie",
  "departement",
  uai,
  "age",
  "age" as tranche,
  case
    when derniere_classe = '24' then null
    when derniere_classe::integer > 5 then 6
    else derniere_classe::integer
  end as derniere_classe,
  retard
from
  v_1718_retard_tmp_int
where
  "age" between 12 and 15
  or "age" is null
```

ORACLE

→

POSTGRESQL

On peut voir sur l'exemple, que sous **PostgreSQL** il faut expliciter au maximum les types de chaque champ sélectionné, alors que sous **ORACLE** il convertit directement et automatiquement (les champs de toutes les tables sont de type varchar (chaîne de caractère) par défaut).

La plus grande partie du temps, j'ai dû travailler avec des expressions conditionnelles, je vais vous illustrer avec des exemples tout en expliquant le mode de fonctionnement :

COALESCE :

```
COALESCE(crel_saisie_2011.nb_dispositif_annee::integer, 0) = 0
```

(La fonction COALESCE renvoie le premier de ces arguments qui n'est pas nul.)

CASE :

```
CASE  
  WHEN cre1_saisie_2011.derniere_classe::text = '24'::text THEN NULL::integer  
  WHEN cre1_saisie_2011.derniere_classe::integer > 5 THEN 6  
  ELSE cre1_saisie_2011.derniere_classe::integer  
END AS derniere_classe,
```

(La fonction CASE reprends le principe d'un switch, si le résultat est vrai, alors la valeur de l'expression CASE est le résultat qui suit la condition)

On peut voir sur les exemples que l'on force le changement de type de données des champs des tables grâce à l'attribut '::integer', qui était de base en type chaîne de caractère en type entier, cela a été obligatoire pour le bon fonctionnement de toutes les fonctions, vu qu'elles fonctionnent seulement avec des numériques et non pas des caractères.

Une fois les deux premières parties de la mission principale réalisées avec succès, il nous faut effectuer la dernière étape consistant à adapter le code applicatif PHP, qui était développé pour **ORACLE** et pas toujours adapté à **PostgreSQL**.

Pour cela, dans un fichier différent du code, nous définissons donc toutes les constantes qui serviront d'identifiants pour la connexion à la base de données. ↴

```
// Activation de la base de données
define( "C_DB_ACTIVE", FALSE );           # TRUE pour utiliser une base oracle
define( "C_DB_HOST", "localhost" );       # Localhost
define( "C_DB_USER", "scott" );           # Login et mot de passe de la connexion utilisée (TODO)
define( "C_DB_PASS", "tiger" );          # pour la gestion des accès SQL à la base de données (TODO)
define( "C_DB_PORT", "1521" );           # Port de la base de donnée par défaut
define( "C_DB_SID", "ORCL" );
define( "C_DB_NAME", "(DESCRIPTION=(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = " . C_DB_HOST . ")(PORT = " . C_DB_PORT . "))) (CONNECT_DATA=(SID=" . C_DB_SID . ")))");
```

Après avoir défini les constantes au préalable, nous allons créer et paramétrer le fichier qui contiendra la connexion directe à la base de données en récupérant les constantes :

On commence à écrire le constructeur dans une fonction en ayant récupéré les constantes dans des variables au préalable. ↴

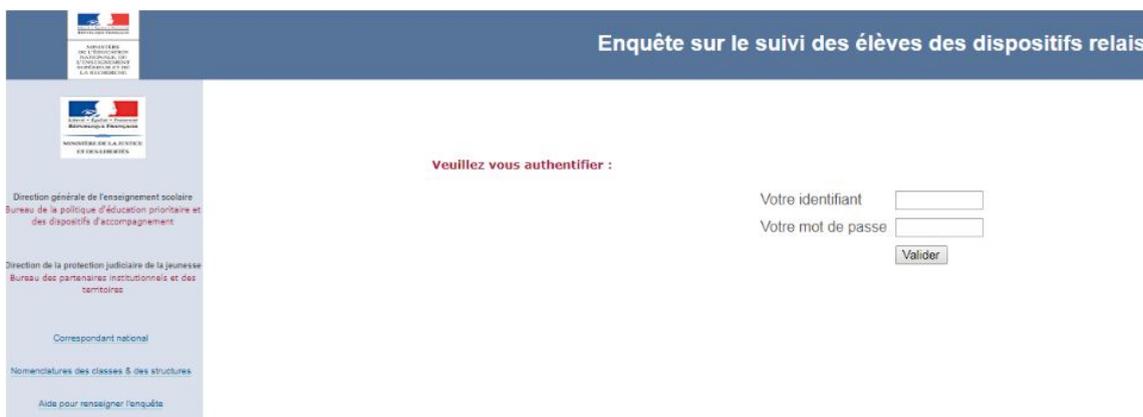
```
function DB($dbHost = "", $dbName = "", $dbUser = "", $dbPass = "", $dbPort = "")
{
    if($dbHost != "") $this->Host = $dbHost;
    //echo "HOST / " . $this->Host . "<br>";
    if($dbName != "") $this->Database = $dbName;
    //echo "DBNAME/ " . $this->Database . "<br>";
    if($dbUser != "") $this->User = $dbUser;
    //echo "USER/ " . $this->User . "<br>";
    if($dbPass != "") $this->Password = $dbPass;
    //echo "PASSWORD/ " . $this->Password . "<br>";
    if($dbPort != "") $this->Port = $dbPort;
    //echo "PORT/ " . $this->Port . "<br>";
}
```

Pour pouvoir se connecter à la base de données et ainsi accéder au portail de l'application web (si la connexion n'est pas sans faute, aucun accès n'est donné), nous avons mis en place la fonction connect() dans

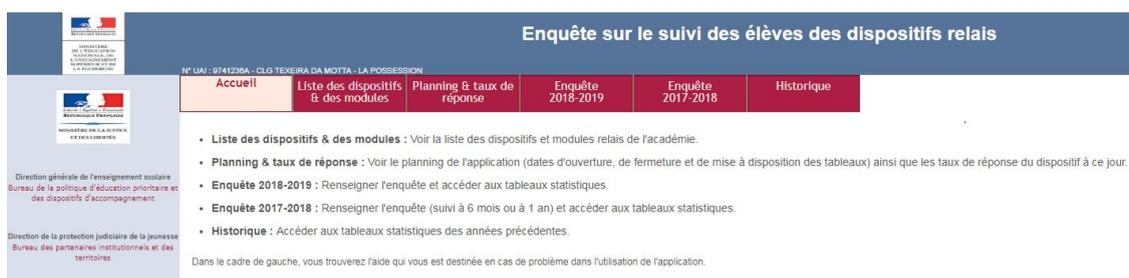
laquelle nous avons inséré la fonction 'pg_connect' vu que nous utilisons actuellement une base de données **PostgreSQL** et en y mettant tous les paramètres initialisés dans le constructeur. ↴

```
function connect()
{
    if($this->link_id == 0)
    {
        $this->link_id = pg_connect("host=$this->host port=$this->port dbname=$this->database user=$this->user password=$this->password");
    }
    if (!$this->link_id)
    {
        $this->halt("link_id == false, connect failed");
    }
    if(pg_last_error( $this->link_id))
    {
        $this->errno = NULL;
        $this->error = pg_last_error($this->link_id);
        $this->halt("Cannot select database <I>".$this->database."</I>");
    }
}
}
```

Une connexion sans soucis de paramétrage et de base de données, nous permet un accès à l'application web du dispositif relais comme ceci ↴



The screenshot shows a login page with a blue header containing the title 'Enquête sur le suivi des élèves des dispositifs relais'. On the left, there is a sidebar with logos and text identifying the 'Direction générale de l'enseignement scolaire' and 'Bureau de la politique d'éducation prioritaire et des dispositifs d'accompagnement'. The main content area features a red heading 'Veuillez vous authentifier :'. Below this, there are two input fields: 'Votre identifiant' and 'Votre mot de passe', followed by a 'Valider' button.



The screenshot shows the main menu of the application. It has a blue header with the title 'Enquête sur le suivi des élèves des dispositifs relais'. Below the header, there is a navigation bar with several menu items: 'Accueil', 'Liste des dispositifs & des modules', 'Planning & taux de réponse', 'Enquête 2018-2019', 'Enquête 2017-2018', and 'Historique'. Below the navigation bar, there is a list of bullet points providing information about the application, such as 'Liste des dispositifs & des modules', 'Planning & taux de réponse', 'Enquête 2018-2019', 'Enquête 2017-2018', and 'Historique'. At the bottom, there is a note: 'Dans le cadre de gauche, vous trouverez l'aide qui vous est destinée en cas de problème dans l'utilisation de l'application.'

Après vérification de l'intégralité de la plateforme, et des performances, certaines requêtes SQL étaient faussées car elles étaient prévues pour **ORACLE**, ce qui entraîne l'inaccessibilité de certaines pages suite à la migration. J'ai donc fait le nécessaire en adaptant le code et les requêtes pour les rendre fonctionnelles.

Concernant les problèmes de performance, nous sommes intervenus en deux étapes :

- Premièrement nous avons fait en sorte d'utiliser les fonctions **Vaccum** et **Analyze** en SQL directement intégrées à **DBeaver**. Celles-ci vont gérer dynamiquement les tables de la base de données et faire en sorte de supprimer les lignes vides, pour libérer de l'espace et ainsi optimiser les ressources.
- Deuxièmement nous avons créé des clés étrangères sur certaines tables (celles où il y a notamment le plus de données) et par la suite créé des index* pour accélérer le temps d'exécution et améliorer les performances.

Petit exemple pour illustrer la création d'un index.

```
CREATE INDEX idx_fk_academie_departement  
ON crel_champ (academie, departement);  
  
CREATE INDEX idx_fk_uai_annee_scolaire  
ON crel_saisie_2011 (numero_uai, annee_scolaire );
```

* : les index sont des ressources très utiles qui permettent d'accéder plus rapidement aux données.

Résultat de la mission

Le projet de migration **Oracle** vers **PostgreSQL** fut une réussite, tout le travail demandé a été réalisé avec succès, aucune perte n'est à déclarer au niveau des tables ou des informations. La migration et l'accès direct à la plateforme en ayant migré notamment les informations contenues dans chacune des tables de la base de données **Oracle** vers

PostgreSQL et l'adaptation du code applicatif PHP pour qu'il fonctionne avec la base de données de cette dernière, a correctement été réalisé.

Enseignement du stage

Je suis très content du travail abouti, au fil des semaines j'ai appris de nouvelles choses, j'ai compris notamment certains points qui me paraissaient flous.

J'ai appris ce qu'étaient les expressions conditionnelles citées dans la partie '**Extrait de code et explications**', à quoi elles servent, leur fonctionnement ainsi que leur importance dans la création des fonctions directement dans la base de données en langage SQL.

Cette expérience m'a aussi permis de répondre aux questions que j'avais en ce qui concerne les moyens utilisés par les entreprises et développeurs.

De plus, j'ai pu assister à une présentation **GitLab*** avec tous les développeurs du service. Elle avait pour but de montrer les différents aspects du système : partager des projets et leur documentation avec d'autres services du Ministère, permettre l'optimisation d'envoi de projet à la production ou bien pour un travail collectif. On m'a même demandé mon avis sur **GitLab**, les avantages à en tirer pendant la réunion.

* : GitLab est un logiciel libre permettant un système de suivi des bugs, l'intégration continue et la livraison continue de projet de développement.

Difficultés rencontrées

J'ai été confronté à plusieurs difficultés pendant le stage, surtout au début, car je ne pouvais pas juste modifier les lignes de code pour que l'application fonctionne sous **PostgreSQL**, je devais modifier toute la

syntaxe SQL, ce qui m'a pris du temps et j'ai réussi à faire face et à comprendre à un moment donné, avec l'aide de ma tutrice de stage et des collègues du service.

J'ai également dû faire face à un souci inattendu, la dernière journée de stage alors que tout le travail demandé était terminé avec succès, en voulant nettoyer une table proprement, la base de données ne se comportait plus correctement, j'ai dû réajuster toutes les clés étrangères et primaires, les vues, et nettoyer des espaces vides des champs et des tables pour que la base de données refonctionne correctement.

Conclusion

Le Ministère de l'Éducation Nationale et de la Jeunesse a été satisfait du travail que j'ai fourni et de la volonté de faire ce qui a été demandé et je suis très fier d'avoir pu contribuer et participer à l'évolution de l'application web en question.

Ce stage a répondu à mes attentes et aux questions sur mon orientation future en tant que développeur web et plus particulièrement vers les applications web métiers.

Enfin, je tiens à exprimer ma satisfaction d'avoir pu travailler dans de bonnes conditions tant au niveau matériel qu'au niveau humain.