

# Compte-Rendu d'Activité

Projet concerné : Slave Narratives - Groupe 2

## A) Contexte

Lors de l'arrivée des alternants en renfort, le projet était déjà très bien avancé, à tel point que la majorité des fonctionnalités demandées étaient déjà réalisées. En attendant que des tâches soient trouvées, nous avons pu prendre en main le projet pendant les premiers jours, car il était tout de même d'une certaine ampleur. De plus, j'avais personnellement des rendez-vous le jeudi de la première semaine. Ainsi, cette première semaine de projet n'a pas été très productive de mon côté, puisque j'ai réellement pu commencer mon travail le vendredi.

## B) Mon rôle

De par l'absence de fonctionnalités à ajouter, je me suis attelé à une tâche que nous avons jugée importante, à savoir le refactoring de la base de données. Il s'agissait encore de la base initiale du projet, qui était fournie avec le site web déjà existant. Celle-ci présentait de nombreux défauts, comme le manque de clarté et l'incohérence des noms des tables et des attributs. Elle causait de nombreuses incompréhensions lorsqu'elle était utilisée dans le code, et nous avons trouvé judicieux de la rendre plus lisible pour permettre une meilleure maintenabilité du projet dans le futur.

L'issue GitHub associée est disponible [ici](#).

## C) Travail réalisé

Le refactoring de la base de données allait potentiellement impliquer d'importants changements dans le code. Ainsi, pour simplifier mon travail et éviter les conflits à chaque changement temporaire de la base de données, j'ai préféré créer une nouvelle branche git appelée « refactor-db » pour isoler ces modifications (branche disponible [ici](#)).

La première chose à faire était d'analyser la base de données existante et surtout de comprendre le sens de chaque table et de chaque attribut. J'ai pu ressortir de cette analyse deux constats :

- Aucun changement majeur impliquant de changer le schéma global de la base n'était nécessaire ;
- Les noms des tables et attributs étaient plus que douteux, et certains éléments semblaient inutilisés.

Le premier constat était positif, car mon travail n'allait pas impliquer de lourds changements dans le code.

En concertation avec les autres membres de l'équipe, j'ai pu identifier le sens de quasiment toutes les tables et de tous les attributs. J'ai donc pu établir un premier dictionnaire des données, qui serait la base du refactoring nécessaire. La plupart des éléments dont je n'avais pas compris le sens étaient en réalité inutilisés. On retrouvait notamment trois tables différentes qui indiquaient la même chose : des points sur la carte. Une s'appelait simplement « points », une s'appelait « fullpt\_1 », et la dernière « pts\_publication ». Des noms qui n'allaient évidemment pas m'aider à identifier la nuance entre chacune de ces tables. En réalité, la solution était plus simple car seule la table « points » était réellement utilisée. Autre problème notable, chaque table représentant un point ou une zone géographique, stockait deux fois la même donnée sous deux formes différentes.

Pour les nouveaux noms des tables et attributs, nous étions tous d'accord qu'il était plus approprié de les renommer en anglais. Je me suis donc occupé de les renommer, et de mettre à jour les endroits dans le code où ils étaient utilisés. Cela a par la suite créé des conflits avec le développement effectué par l'équipe en parallèle. J'ai résolu les conflits directs lors de la pull request de ma branche « refactor-db » vers la branche « main ». Malgré quelques oublis de ma part, la modification du code s'est bien passée, et n'a causé aucune casse.

L'analyse complète de la base de données initiale, comprenant également les changements effectués, est disponible [ici](#). Un exemple de modification d'une table se trouve en annexes.

A la demande de certains membres de l'équipe, je me suis également penché sur un éventuel non-sens des attributs stockés dans la table « author ». En effet, suite à un quiproquo lors d'une précédente réunion, il leur semblait que certains attributs de cette table ne soient pas à la bonne place, et seraient plus adaptés dans une table « slave » à part. Après avoir analysé le contenu de la table « author », j'ai identifié deux cas possibles :

- Tous les auteurs sont des esclaves ;
- Chaque donnée concerne un auteur et l'esclave dont il fait le récit.

Après contact avec la cliente, il s'est avéré que nous étions dans le premier cas, qui n'impliquait donc pas de changement dans la base de données. Cependant, cela nous a permis de redéfinir certains termes, et nous avons renommé la table « author » en « narrator ».

## **D) Et ensuite ?**

Ce refactoring de base de données aura un impact important pour la suite. Tout d'abord, il permettra à toute personne travaillant sur le projet dans le futur de comprendre rapidement sa structure, et sera également aidé par le dictionnaire des données réalisé. Ensuite, ce travail a également révélé la nécessité de retravailler l'entièreté du code d'origine, avec notamment de nombreux noms de variables incohérents et en plus en français. C'est un travail qui a commencé le dernier jour de projet pour les alternants, je n'aurai donc pas la possibilité d'y participer.

Pour conclure, ce travail d'amélioration de la base de données, et bientôt du code de manière générale, permettra je l'espère de simplifier l'amélioration du site par la suite et de permettre à Madame Baduel d'y ajouter plus rapidement de nouvelles fonctionnalités !

## Annexe 1 : Exemple « avant » (tab\_auteurs)

### Table TAB\_AUTEURS

#### Description :

Table contenant les informations sur les narrateurs.

#### Actuellement :

Nom	Type	Description
nom	varchar(58)	Nom narrateur
naissance	varchar(27)	Année de naissance
lieu_naissance	varchar(38)	Lieu naissance
deces	varchar(10)	Année de mort
lieu_deces	varchar(29)	Lieu de mort
lieu_esclavage	varchar(53)	Lieu d'esclavage
moyen_lib	varchar(117)	Moyen de libération
lieuvie_ap_lib	varchar(72)	Lieu de vie après libération
origine_parents	varchar(164)	Origines des parents de l'esclave
militant_abolitionniste	varchar(37)	Est un abolitionniste
particularites	varchar(113)	Particularités
plrs_recits	varchar(3)	Si il a écrit plusieurs récits
id_narrateur	varchar(3)	ID narrateur
op_source	varchar(164)	?

## Annexe 2 : Exemple « après » (narrator)

### Table narrator

#### Description

Table contenant les informations des narrateurs de récits.

#### Structure

Nom	Type	Description
id_narrator	int(11)	ID narrateur
name	varchar(64)	Nom
birth	varchar(32)	Année naissance
birth_place	varchar(64)	Lieu de naissance
death	varchar(32)	Année de mort
death_place	varchar(64)	Lieu de mort
enslavement_place	varchar(64)	Lieu d'esclavage
freeing_ways	varchar(128)	Moyen de libération
living_place_after_freeing	varchar(128)	Lieu de vie après libération
parents_origin	varchar(256)	Origines des parents
abolitionist	varchar(64)	Est un abolitionniste
particularities	varchar(128)	Particularités
has_wrote_several_narratives	tinyint(1)	Si il a écrit plusieurs récits