CY CERGY PARIS UNIVERSITE

RAPPORT

Projet d'intégration

Licence d'Informatique troisième année

Sur le sujet

Student Tools : PostgreSQL GUI

Construction d'un outil de visualisation graphique d'une base de données PostgreSQL

Rédigé par

Groupe composé de :

MEGBLETO Carnel et

MAKITA-BOUNGOU Andreich-Vertus

2020 – 2021

SOMMAIRE

REMERCIEMENTS

Nous remercions particulièrement Mr. Marc LEMAIRE d'avoir proposé ce projet d'intégration et de nous avoir accompagné dans sa réalisation.

Nos remerciements vont aussi à tout le corps professoral du Département Informatique pour leur implication à la réussite de cette année si particulière ainsi qu'à tous les étudiants de la Licence 3 Informatique 2020-2021.

LISTES DES FIGURES

INTRODUCTION

Ce présent rapport a pour but de présenter les différentes étapes qui ont mené à la réalisation de notre projet. Il donne une vue assez détaillée sur le projet notamment son but, la répartition des tâches, les outils utilisés pour la conception ainsi que son fonctionnement et enfin les objectifs atteints.

I. Aperçu du projet

I.1. Objectif du projet

Conception et réalisation d'un outil de visualisation de base de données PostgreSQL qui permettra aux utilisateurs de facilement se connecter à une base de données PostgreSQL de leur choix afin de visualiser sa structure et pouvoir identifier les clés primaires, les clés étrangères ainsi que les relations entre différentes tables de provenance.

I.1.1. Définition de PostgreSQL

PostgreSQL est un système de gestion de bases de données relationnelles objet fondé sur POSTGRES. Ce dernier a été développé à l'université de Californie au département informatique de Berkeley.

PostgreSQL est un descendant libre du code original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes :

- Requêtes complexes ;
- Clés étrangères ;
- Triggers ;
- Vues modifiables ;
- Intégrité transactionnelle ;
- Contrôle des versions concurrentes (MVCC, acronyme de « MultiVersion Concurrency Control »).

De plus, PostgreSQL peut être étendu par l'utilisateur de multiples façons, en ajoutant, par exemple :

- De nouveaux types de données ;
- De nouvelles fonctions ;
- De nouveaux opérateurs ;

- De nouvelles fonctions d'agrégat ;
- De nouvelles méthodes d'indexage ;
- De nouveaux langages de procédure.

Et grâce à sa licence libérale, PostgreSQL peut être utilisé, modifié et distribué librement, quel que soit le but visé, qu'il soit privé, commercial ou académique.

I.2. Organisation autour de la réalisation projet

Durant le déroulement du projet nous avons eu plusieurs rencontres avec Monsieur Marc Lemaire afin de faire un point d'avancement.

I.2.1. Mise au point sur le projet

a. SEMAINE 1 &2 :

Point d'avancement avec Monsieur Marc Lemaire.

- Prises de connaissance du cahier des charges :
- Identification des besoins de l'application ainsi que des attendues.

b. SEMAINE 3 & 4 :

Point d'avancement avec Monsieur Marc Lemaire.

- Réalisation d'une maquette du produit fini avec **figma** afin de mieux pouvoir faire des choix en ce qui concerne les technologies, librairies et langages de programmation que nous aurons à utiliser.
- Choix définitif des technologies à utiliser et début de la conception du backend de notre application.

c. SEMAINE 5&6 :

Point d'avancement avec Monsieur Marc Lemaire.

- Début de la conception du frontend de notre application en nous servant de l'api que nous avions créé au niveau de notre backend.
- Début de la rédaction de notre rapport de projet

d. SEMAINE 7&8 :

Fin de la réalisation du frontend et du backend de notre application

- Finalisation **de** la rédaction de notre rapport de projet.
- Réalisation des slides pour notre présentation.

I.2.2. Répartition des tâches

a. La partie technique :

- Réalisation d'une maquette de la plateforme finale avec l'outil Figma **(Vertus)**
- Construction de l'API et formulation des requêtes qui nous permettraient de questionner la base de données pour obtenir l'information recherchée**. (Carnel et Vertus)**
- Identification de la libraire qui nous permettra d'effectuer de façon adéquates les connections entre les tables**. (Vertus)**
- Consommation et utilisation de l'information produit par l'API en effectuant des requêtes http au niveau de notre frontend. **(Carnel)**
- Réalisation du site avec React-JSX et CSS pour le styling. **(Carnel)**

b. La partie documentation :

- Rédaction du rapport **(Vertus)**
- Conception du PowerPoint de la présentation **(Carnel)**
- Réalisation de la vidéo de démonstration **(Carnel et Vertus)**

II. Technologies et langage de programmation utilisées

La solution technique choisie pour la conception de notre outil de visualisation est orientée web et entièrement basée sur du JavaScript.

II.1. JavaScript

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. Elle est la troisième couche des technologies standards du web, les deux premières étant HTML et CSS

Le choix de s'appuyer sur ce langage c'est pour la diversité de ses librairies et de ses Framework.

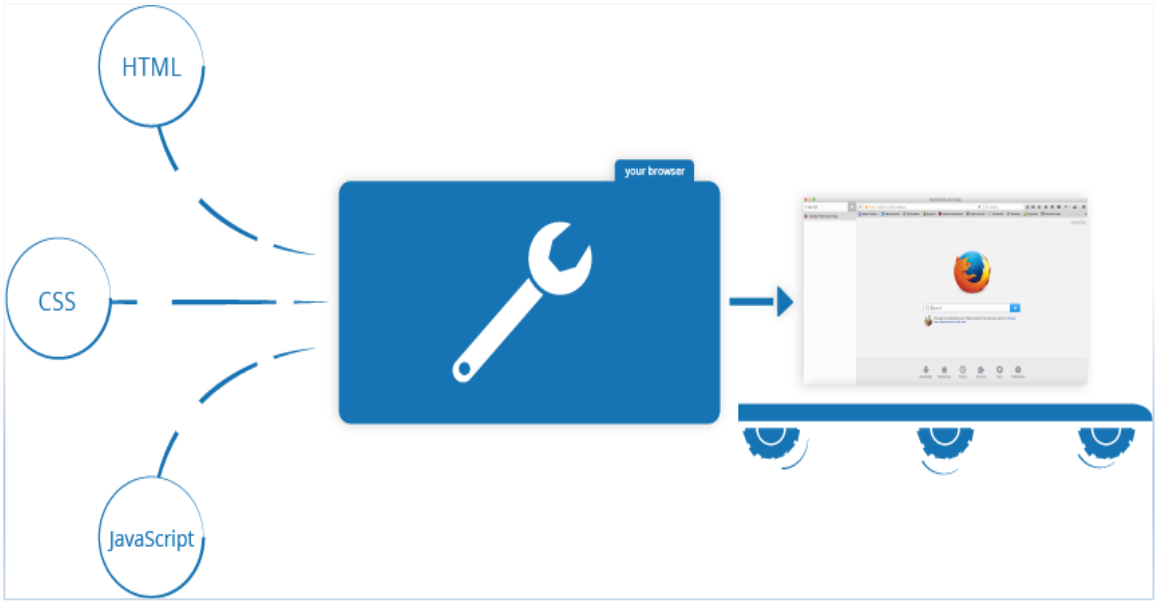


Figure II-1: JavaScript

Nous avons également utilisé CSS pour ce qui de la mise-en forme de notre plateforme.

II.1.1. Node.Js

Node.js est une plateforme logicielle libre en JavaScript, qui permet de développer rapidement un web service répondant à des données formatées ou pages internet.

Elle utilise la machine virtuelle V8, la librairie libuv pour sa boucle d'évènements, et implémente sous licence MIT les spécifications Commons.

Les grands principes sous-tendant Node.js sont :

- Exécution pilotée par les événements
- Appels de fonctions asynchrones (utilisation de callback)
- Entrées/sorties non bloquantes

Dans le cadre de notre projet :

__Javascript__ étant un langage de Scripting rendu côté __client__ nous avions eu besoin du Framework __node.js__ pour pouvoir l'utiliser __côté server__ un peu comme PHP qui est un langage de Scripting rendue côté server.

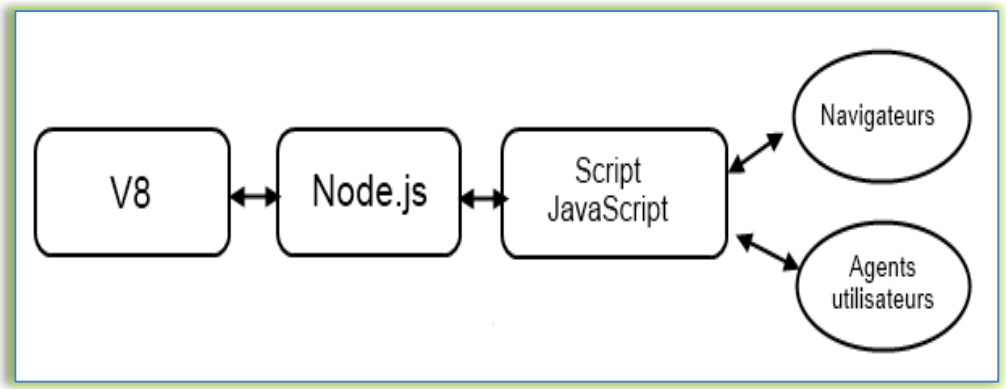


Figure II-2: Node JS

II.1.2. Express.Js

Express.Js est le Framework standard pour la conception d'application Web pour Node.js, il a un développement minimal, flexible et rapide. Il est le Framework de choix pour la conception de server et fournit un ensemble robuste de fonctionnalités pour développer des applications Web et mobiles.

II.1.3. React.Js

React.Js est une librairie JavaScript open source dédiée à l'écriture d'interfaces utilisateurs, propose une approche bien particulière sur la manière de définir une vue. Elle est orientée composants.

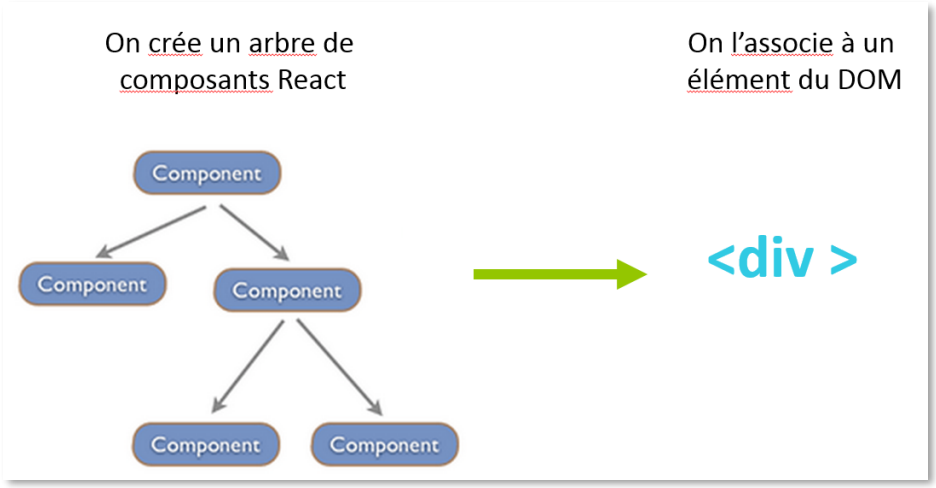


Figure II-3: React.JS

II.1.4. React X-arrows

Librairie javascript écrite en type Script qui :

Connecte des éléments d'une page web entre eux en prenant comme paramètre un id ou une référence pour marquer le point de départ de la flèche ainsi que le point d'arriver. (Pas besoin que l'utilisateur de notre solution l'installe manuellement elle est automatiquement intégrée dans le fichier package. Son du dossier frontend de notre application).

Mode d'utilisation :

```
<Xarrow
      start={box1Ref} //can be react ref
      end="elem2" //or an id
    />
```

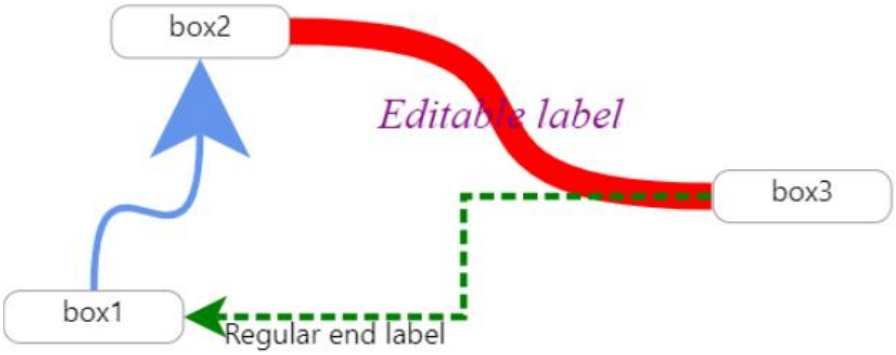


Figure II-4: Libraire connection

II.1.5. React -Router

React Router est une bibliothèque de routage (routing) standard dans **React**. Cela rend l'interface de l'application synchrone avec l'**URL** du navigateur. Le **React Router** vous permet de router clairement le "flux de données" (data flow) dans votre application.

Dans le context de notre application nous avions utiliser react-router pour permettre a l'utilisateur de notre application de pouvoir naviguer d'une page a l'autre sans que nous ayons a recharger la page ou a le rediriger.

Ici l'utilisateur peut naviguer de la page view a table a la page View Relationship Grâce a react-router.

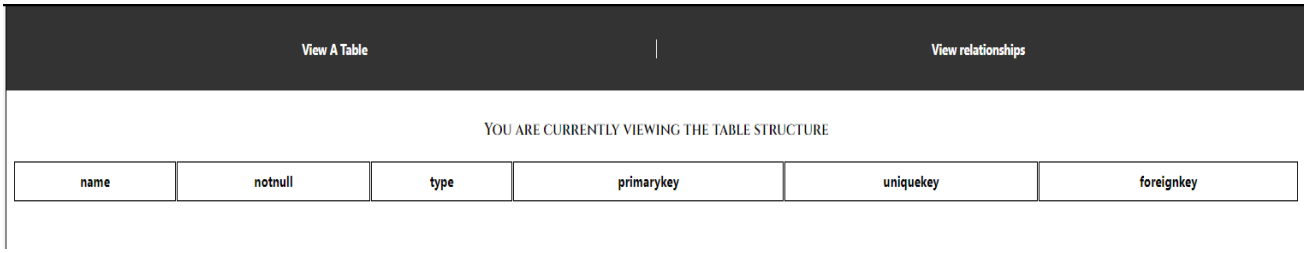


Figure II-5: React-Router Implémentation

II.1.6. Nodemon

Nodemon est un outil développer en javascript que nous avions utilisé au niveau du backend de notre application et qui est charger de redémarrer notre server backend a chaque fois que

nous effectuons un changement ou qu'un changement au niveau du code est détecté au niveau de notre backend. (L'utilisateur de notre application n'aura pas besoin de l'installer manuellement car il est automatiquement rajouté après avoir installé notre application).

Figure II-6: Installation Nodemon

III. La Conception et la mise en place

III.1. Fonctionnement

a. Comment faire fonctionner le frontend de notre application

Dans le contexte de notre projet pour lancer le frontend de notre application il faut :

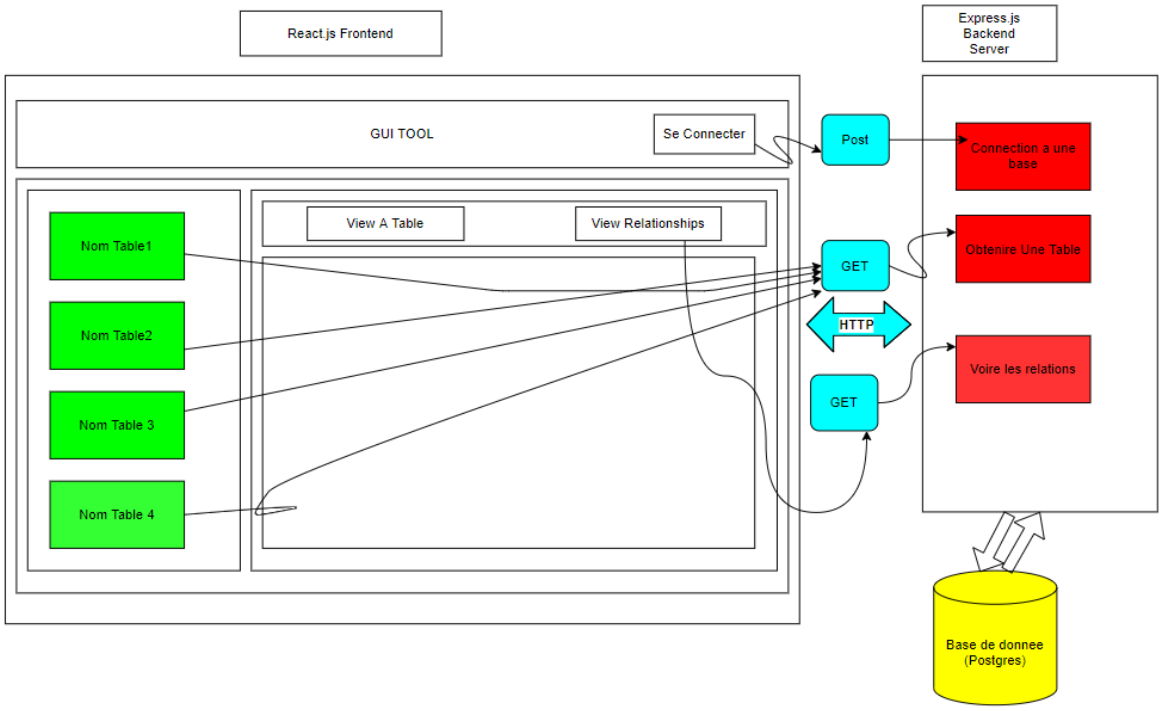
1. S'assurer que node.js est installer sur la machine
2. Faire *: **cd frontend*** depuis le root directory de notre application pour accéder au dossier contenant les informations relatives à notre application.
3. **Faire : *npm Install*** pour que le Node **P**ackage **M**anager puisse installer toute les dépendances et librairies indispensable au bon fonctionnement du frontend de notre application
4. Faire : ***npm start*** et attendre que le server démarre.

b. Comment lancer le backend de notre application

Dans le cadre de notre projet pour lancer le backend de notre application il faut :

1. 1-S'assurer que node.js est installer sur la machine
2. 2- Faire : ***cd back*** depuis le root directory de notre application pour accéder au dossier contenant les fichiers relatifs au backend de notre application
3. 3- Faire : ***npm Install*** pour que le Node **P**ackage **M**anager Install toutes les dépendances et package indispensable au bon fonctionnement de notre application
4. 4- Faire : ***npm run dev*** et attendre que notre server démarre.

III.2. Architecture de l'application



Bouton se connecter :

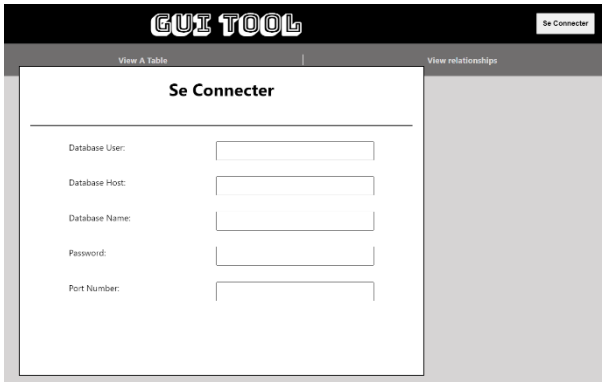


Figure III-1: Architecture du fonctionnement

En cliquant sur ce bouton l'utilisateur est prié de remplir les informations indispensables à la connexion a une base de données PostgreSQL tels que :

- **Le nom d'utilisateur**
- **L'hôte**
- **Le nom de la base de données** à laquelle il souhaite se connecter.
- **Le mot de passe** associer à la base de données à laquelle il souhaite se connecter
- **Le numéro de port** sur lequel tourne la base de données PostgreSQL.

NB : Pas de Button submit au niveau du formulaire de connexion vue que les informations qui y sont renseigner son automatique renvoyé au serveur en temps réel.

a. Plateforme

Reste de la platform, Buton (View A Table & view relationships)

Apres avoir renseigner les informations relatives à la connexion à la base de données le reste de la platform est automatiquement générer de façon dynamique en fonction de la base à laquelle l'utilisateur est connecté.

b. Barre de navigation

Cela vaut le cou pour :

- La barre de navigation qui est générer de façon dynamique pour afficher le nom des tables présentes dans la base de données à laquelle l'utilisateur est présentement connecté et qui sert également pour déclencher des requêtes http a chaque fois que l'utilisateur clique sur un nom de table.
- Les tables et les connections entre elle sont générées de façon dynamique en fonction de la structure de la base de données à laquelle l'utilisateur est connecté.

IV. Les objectifs atteints

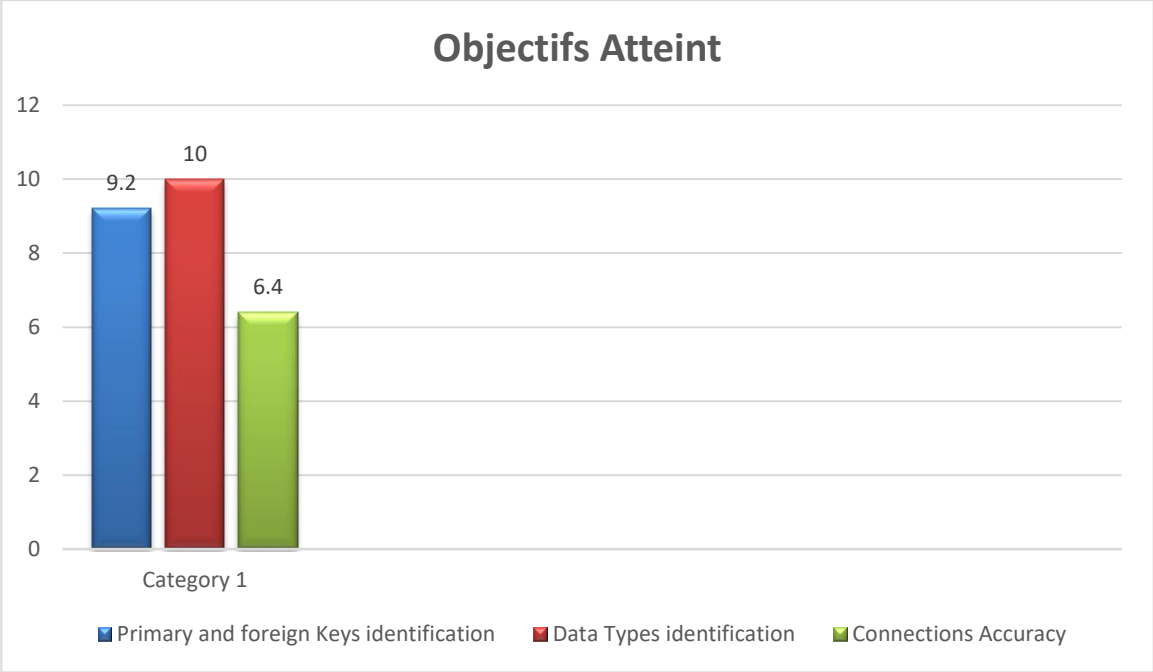


Figure IV-1: Graphes d'objectifs

a. Primary and foreign Keys identification :

Nous avions donnée 9.2/10 a l'habileté de notre solution à identifier les clés primaires et étrangères car les données se répète dans les tables lorsqu'un attribut est à la fois une clé primaire et une clé étrangère.

b. Attribues types identification :

Nous avions donnée 10/10 a l'habileté de notre solution à identifie le type des attributs présents dans les tables de la base de données.

c. Connections accuracy :

Nous avions donnée 6.4/10 a l'habileté de notre solution à faire des connections entre les attributs présents dans les tables et cela est due au fait que la librairie que nous avions utilisée

(React x-arrows) ne dispose d'aucun moyen de distinguer les cas où les attributs de départ et d'arriver on le même nom.

CONCLUSION

Ce projet, nous a permis de développer une solution répondant au besoin existant des utilisateurs de base de données Postgres qui est de pouvoir identifier de façon claire, rapide et précise la structure d'une base de données à travers une description de ces clés primaires, étrangères ainsi que des relations entre les diffèrentes tables.

TABLES DES MATIÈRES