

# Implémentation d'une application web permettant de passer des requêtes paramétrées et de visualiser les réponses, en utilisant Django+PostgreSQL

(Projet BDDR, 2023-2024)

## Contexte:

L'affaire Enron ([https://fr.wikipedia.org/wiki/Scandale\\_Enron](https://fr.wikipedia.org/wiki/Scandale_Enron)) est connue comme l'un des plus grands cas de manipulation financière aux États-Unis. Parmi les diverses enquêtes judiciaires menées, la fouille des contenus des e-mails a constitué une piste essentielle pour comprendre la dynamique entre les différents acteurs et leur rôle dans l'affaire.

## Sujet:

Il s'agit de développer une application destinée aux enquêteurs (non informaticiens). Elle doit leur proposer un jeu de formulaires leur permettant d'explorer, de visualiser et d'analyser les données. Un formulaire correspond à une interrogation/visualisation, il doit permettre à l'enquêteur de fixer différents paramètres de la requête ainsi que d'éventuelles options pour la visualisation. Il est nécessaire ici de se glisser dans la peau d'un enquêteur pour imaginer quelles requêtes sont pertinentes, et quelles visualisations sont judicieuses. Toute réalisation allant au-delà du minimum requis se traduira par des points de bonus au niveau de la note finale.

Les données sont une arborescence de boîtes mails et un fichier XML. L'archive est récupérable depuis <https://math.univ-angers.fr/perso/jaclin/enron/>.

### Socle minimal des requêtes :

1. attributs d'un employé (identité, catégorie, e-mails) déterminé par son nom ou par l'une de ses adresses e-mail ;
2. employés ayant envoyé et/ou reçu plus que (resp. moins que) x mails dans un intervalle de temps choisi, avec possibilité de faire la différence entre échanges internes et/ou internes-externes ;
3. liste des employés ayant communiqué avec un employé donné sur un intervalle de temps donné ;
4. couples d'employés ayant communiqué dans un intervalle de temps choisi (liste ordonnée suivant le nombre de mails échangés, tronquée au-dessous d'un seuil paramétrable) ;
5. jours dans une période donnée ayant connu le plus grand nombre d'échanges de mails (en différenciant les échanges internes des échanges interne-externe) ;
6. mails contenant une liste de mots déterminés, avec diverses présentations (par expéditeur, par destinataire, etc.) et possibilité de visualiser le contenu d'un mail donné ;
7. contenu d'une conversation (déterminer ce qu'est une conversation) : afficher les mails la composant, avec possibilité d'obtenir le contenu d'un mail donné.

Au minimum, 4 requêtes doivent être intégralement faites en SQL.

Toute requête supplémentaire est bienvenue. Proposer, suivant les requêtes, un résultat sous forme de listes et/ou de graphiques (camemberts, histogrammes, etc.).

### Aspects techniques:

Votre application doit être développée avec le framework web **Django** et le SGBD **PostgreSQL**. Pour le peuplement de la base de données, l'utilisation de Python est conseillée, mais vous pouvez choisir une autre technologie. La gestion des versions avec **Git** (sur Github) est obligatoire, ainsi que des mises à jour fréquentes (1 par semaine, au minimum).

La base de données doit être peuplée avec la totalité des données valides fournies, et l'application doit fonctionner avec ce jeu de données complet en un temps raisonnable. En particulier, la démo, à réaliser lors de la soutenance, devra se faire sur la totalité du jeu de données stocké dans la base de données.

### Modalité du projet:

- Projet à réaliser en binôme.
- On considère comme livrables de votre projet :
  - a. le code développé pour l'application ;
  - b. le script de peuplement de la base de données ;
  - c. le script de création de la base de données ;
  - d. le schéma du modèle de la base de données (en format PDF).
  - e. le rapport final ;
  - f. un fichier README pour indiquer comment lancer votre projet dans une autre machine.
- Pour le rapport :
  - a. 10 pages max ;
  - b. trois parties bien définies : contexte, base de données et application ;
  - c. les requêtes SQL exécutées dans votre application.

### Evaluation:

La note est déterminée par la soutenance de projet, au cours de laquelle on évaluera principalement :

- la maîtrise de l'application que vous avez développée ;
- vos connaissances sur le langage SQL : attention, suivant le schéma de données proposé, on vous demandera éventuellement d'ébaucher une requête SQL quelconque ;
- votre compréhension du jeu de données, et de son potentiel d'exploitation.

**Calendrier:**

Semaine	Activité(s)
6	<ul style="list-style-type: none"><li>• renseigner la composition de son binôme et l'URL du repo git sur <a href="https://codimd.math.cnrs.fr/Ddz5J6m6R16K_QvRO1h9JA?view">https://codimd.math.cnrs.fr/Ddz5J6m6R16K_QvRO1h9JA?view</a></li><li>• donner les droits de consultation du repo git aux correcteurs</li><li>• mise en place de l'environnement de développement</li></ul>
6 ~ 9	<ul style="list-style-type: none"><li>• étude des données, nettoyage</li><li>• conception du modèle de données</li></ul>
10	<ul style="list-style-type: none"><li>• point d'étape : chaque binôme expose son modèle de données (le schéma et le script doivent être à jour)</li></ul>
11 ~ 14	<ul style="list-style-type: none"><li>• peuplement de la base de données</li></ul>
15	<ul style="list-style-type: none"><li>• évaluation des scripts de peuplement à partir du dépôt git</li></ul>
16 ~ 19	<ul style="list-style-type: none"><li>• développement de l'application web</li><li>• rédaction du rapport synthétisant les idées suivies, les idées non suivies, les difficultés, les surprises, et le travail réalisé.</li></ul>
20	<ul style="list-style-type: none"><li>• vérifier que le dépôt Git est complet (code + scripts + schéma)</li><li>• <b>Chaque membre du binôme</b> dépose le rapport dans Moodle.</li><li>• Attention: Il ne faut pas déposer les données!</li><li>• Soutenance</li></ul>