

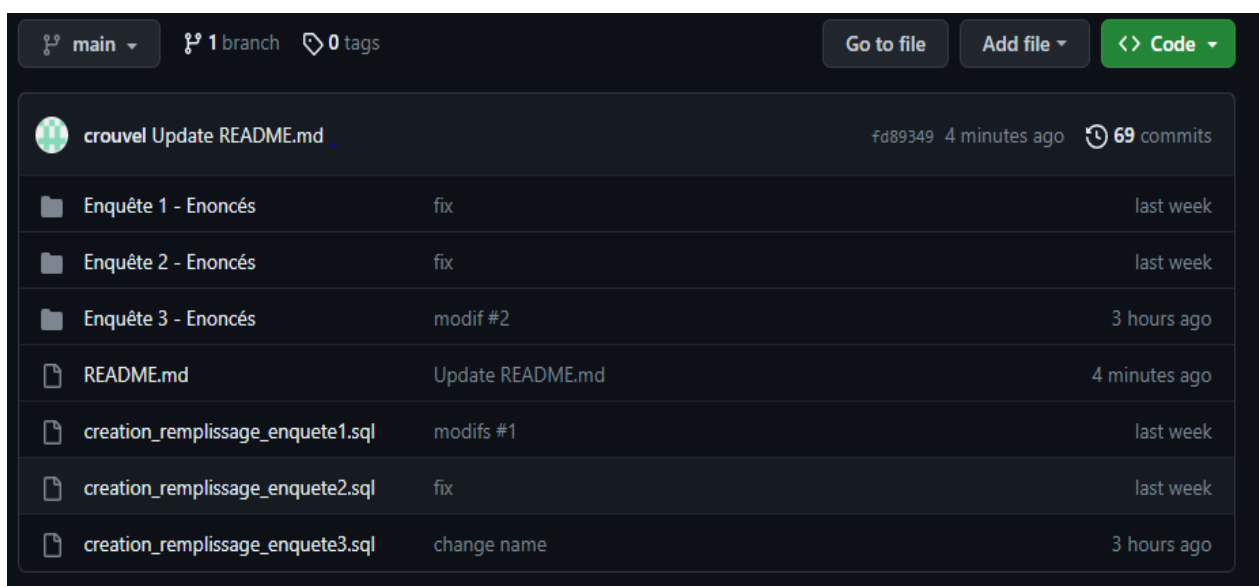
Guide utilisateur

Encadrants

Ce document est destiné aux personnes encadrant la résolution d'enquête(s) SQL. Ici, nous allons expliquer les différents points importants afin d'accompagner les étudiants et de mener à bien ces travaux de programmation.

1. Explication du dossier d'enquêtes

Le dossier Github d'enquêtes se présente ainsi.



Pour les trois enquêtes, un répertoire github regroupe les trois énoncés et les scripts sql associés.

Pour chacune des enquêtes, un dossier contenant les énoncés, avec et sans solutions, est présent. Ils sont disponibles au format pdf et docx. Un fichier possède l'énoncé avec solution, et un autre fichier seulement l'énoncé, destiné à ceux qui devront pratiquer l'exercice.

Les différents niveaux attribués sont : collègue, étudiant non-info et étudiant info. Le niveau augmente graduellement avec le numéro de l'enquête. L'enquête 1 correspond au niveau collègue, l'enquête 2 niveau étudiant non-info, et enquête 3 niveau étudiant info. **C'est donc à vous de choisir le niveau de difficulté adéquat pour les élèves.**

Il y a ensuite les trois fichiers de scripts sql lancés automatiquement dans chaque terminal d'enquête. Les noms sont relativement explicites. Grâce à eux, les tables de données sont créées et pré-remplies au chargement des trois terminaux Crunchy Data.

```
creation_remplissage_enquete1  
creation_remplissage_enquete2  
creation_remplissage_enquete3
```

Si vous souhaitez faire part d'une modification ou reprendre les scripts, ceux sont fichiers dont il sera question.

Le readMe correspond au fichier de documentation du répertoire Github, que vous avez lu auparavant.

2. Points importants à comprendre et communiquer aux élèves

Pour chacune des enquêtes à résoudre, un lien de terminal Crunchy Data est associé. Il faut comprendre que celui-ci est formé de l'URL du terminal en ligne de Crunchy Data et de l'URL qui pointe vers les fichiers au format raw des scripts de création de tables et d'insertion de données, présents dans le dossier github correspondant.

- **Premier point important :**

Faites en sorte que les étudiants n'essaie pas de retracer les lignes de données en voulant accéder directement au répertoire github contenant les scripts en postgresQL. Il faut néanmoins savoir que les volumes de données restent conséquents et ne leur facilitent pas la tâche.

- **Deuxième point important :**

Lorsque vous cliquez sur l'un des liens pour résoudre des enquêtes Crunchy Data, il faut attendre que toutes les lignes de scripts soient exécutées, et que vous puissiez taper des requêtes sql. Laissez les scripts se lancer automatiquement, puis les élèves pourront commencer à travailler lorsque le terminal affichera tout à la fin :

```
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 273
INSERT 0 42
INSERT 0 13154
postgres=#
```

C'est le : postgres=# qui permet de rédiger des requêtes SQL.

- **Troisième point important :**

La syntaxe des requêtes du terminal de Crunchy Data doit respecter des contraintes particulières.

Toutes les commandes basiques de PostgreSQL (SELECT, WHERE, JOIN, AND, ETC) doivent toutes être écrites en **majuscule**, sinon, elles ne s'exécutent jamais.

Pour chaque fin de requête écrite, **un point-virgule doit être indiqué**.

Ceci ne prend pas en compte les sous-requêtes, ou il ne faut pas mettre de point-virgule à la fin de celles-ci.

Ensuite, pour les requêtes écrites sur plusieurs lignes, il faut faire en sorte que les commandes principales soient suivies par un attribut ou mot clé, pour chaque ligne. Vous pouvez voir l'exemple dans les requêtes corrigées des énoncés de correction.