# Mise en œuvre du jeu de test rattaché au document technique du projet OC Pizza

## Contenu de la livraison :

Le jeu de test est constitué de 2 fichiers :

* **Init\_db.sql** : Un script d’initialisation de schéma mysql.
* **Filler.py** : Un script python de génération de données fonctionnelles et de simulation de scénario.  
  (Note : Nous savons que ce script est «Non pep8 compliant» notamment du fait de sa taille, mais la fin justifiant les moyens, nous vous demandons aimablement de ne considérer que le résultat fonctionnel du rendu).

## Prérequis :

* Serveur mysql installé  
  (testé sur : Mysql Community Server 8.0.20)
* Base de donnée fonctionnelle avec un compte ayant les droits nécessaires pour créer un schéma, réceptionner et manipuler le modèle en lecture écriture sur toutes les tables.

## Installation :

Vous pouvez vous placer dans un répertoire dédié et cloner le projet depuis github.   
Ou simplement copier :

- le script «filler.py» situé dans le répertoire «filler» de ce repository .

- le fichier sql «init\_db.sql» d’intialisation de la base situé dans le répertoire «sql» de ce repository.

**Cloner** (mode «ssh»):

# git clone git@github.com:jean-charles-gibier/OC\_pizza.git

**Installer la base et jouer le script init\_db.sql** :

# cd sql

# mysql -u <username> -p < init\_db.sql

**Note importante** : Ce script va créer un schéma nommé ‘**mydb**’ (script directement issu du générateur MysqlWorkbench, libre à vous de modifier les détails d’implantation du schéma, pourvu qu’il reste cohérent avec les instruction du «filler»).

Éventuellement installer un environnement(windows) :

python -m venv <chemin>\venv

venv\Scripts\activate

**Installer les requirements :**

# cd filler

# pip install -r requirements.txt

(installe notamment les packages Faker et MysqlConnector)

**Modifier la configuration dans le script «filler.py» :**

# vim filler.py

# ~

Dans le fichier modifier les éléments de configuration suivants :( «filler.py» ligne 14).

|  |
| --- |
| DB\_USERNAME = "<YOURUSERNAME>" |

|  |
| --- |
| DB\_PASSWORD = "<YOURPASSWD>" |

|  |
| --- |
| DB\_NAME = "<YOURDBNAME>" |

DB\_HOST = "<YOURHOST>"

Modifier également le chemin menant à mysql :  
PATH\_MYSQL = "<chemin vers ...>/bin/mysql.exe"

Et celui menant à notre script de nettoyage :  
SQLCLEAN\_DB = "../sql/truncate\_all.sql"

Sauvegarder(!).

**Lancer le générateur :**

# python3 ./filler.py

Ou (uniquement après un lancement standard)

# python3 ./filler.py ORDERS\_ONLY

**Explications :**

*La première* instruction rempli les tables des entités principales et les tables de liaisons afférentes ( Pizzeria, Employé, Client (Personne), Menu, Recette, Stock, Unité, etc.).   
Elle effectue également le contenu de la 2eme instruction ci dessous.

*La deuxième* instruction effectue un nombre aléatoire de commande sur des pizzerias tirées au hasard, prise en charge par des employés au hasard (en respectant toutefois la relation employé/pizzeria). Cette instruction est utile pour coller à un scénario plus vraisemblable et ne pas partir d’une base «fraîche» (stocks complets) à chaque déclenchement de commande.

**Usage :**

**filler.py [ORDERS\_ONLY]**

**fills all tables of schema OC Pizza with coherent test cases**

**If parameter "ORDERS\_ONLY" is set, the filler will play only**

**the ordering part of the users stories.**

**Tests :**

*Au final (après avoir simulé suffisamment de commandes) des requêtes de test permettent de contrôler les données résultantes, le comportement des relation et la cohérence générale du modèle.*

*# cd sql*

*# vim tests.sql*