



Conception & Modélisation d'un système informatique d'une pâtisserie (El-Foursaane)

Rapport des activités

Moukhtar Ben Ali

28 Mars, 2022



Table des matières

1) INTRODUCTION	2
2) IDENTIFICATION DES ACTEURS	2
3) DIAGRAMME DE CAS D'UTILISATION	3
4) DIAGRAMME DE CLASSE	4
5) CHOIX DE LA TECHNOLOGIE ET CREATION DE LA BASE DE DONNEE	5
6) DIFFICULTES RENCONTREES	9
7) CONCLUSION	10
8) LISTE DES FIGURES	10

INTRODUCTION

Une base de données est un ensemble structuré et organisé de données qui représente un système d'informations sélectionnées de telle sorte qu'elles puissent être consultées par des utilisateurs ou par des programmes. Dans une entreprise, la base de données contient l'ensemble des données concernant les clients, les fournisseurs, les employés, les références des produits fabriqués, etc. Ainsi, dans le cadre de structuration de ses informations.

Le groupe El-Foursaane souhaite réaliser une application web pour la gestion de ses pâtisseries qui se trouve dans la ville de N'Djamena. Pour ce faire, elle propose le cahier des charges dans le document annexe afin de répondre à son besoin :

- *Enregistrement des achats,*
- *Gestion de stock,*
- *Gestion du personnel,*
- *Planification du personnel,*
- *Gestion de la production*

****Travail à Faire : ****

1. Identifier les différents acteurs
2. Réaliser le Diagramme de cas d'utilisation
3. Réaliser le Diagramme de classe
4. Créer votre BD et Implémenter les données nécessaires

Identification des différents acteurs

Aux vue du cahier de charge, voici les acteurs que nous avons pu ressortir :

- Chef de la production
- Administrateur

Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

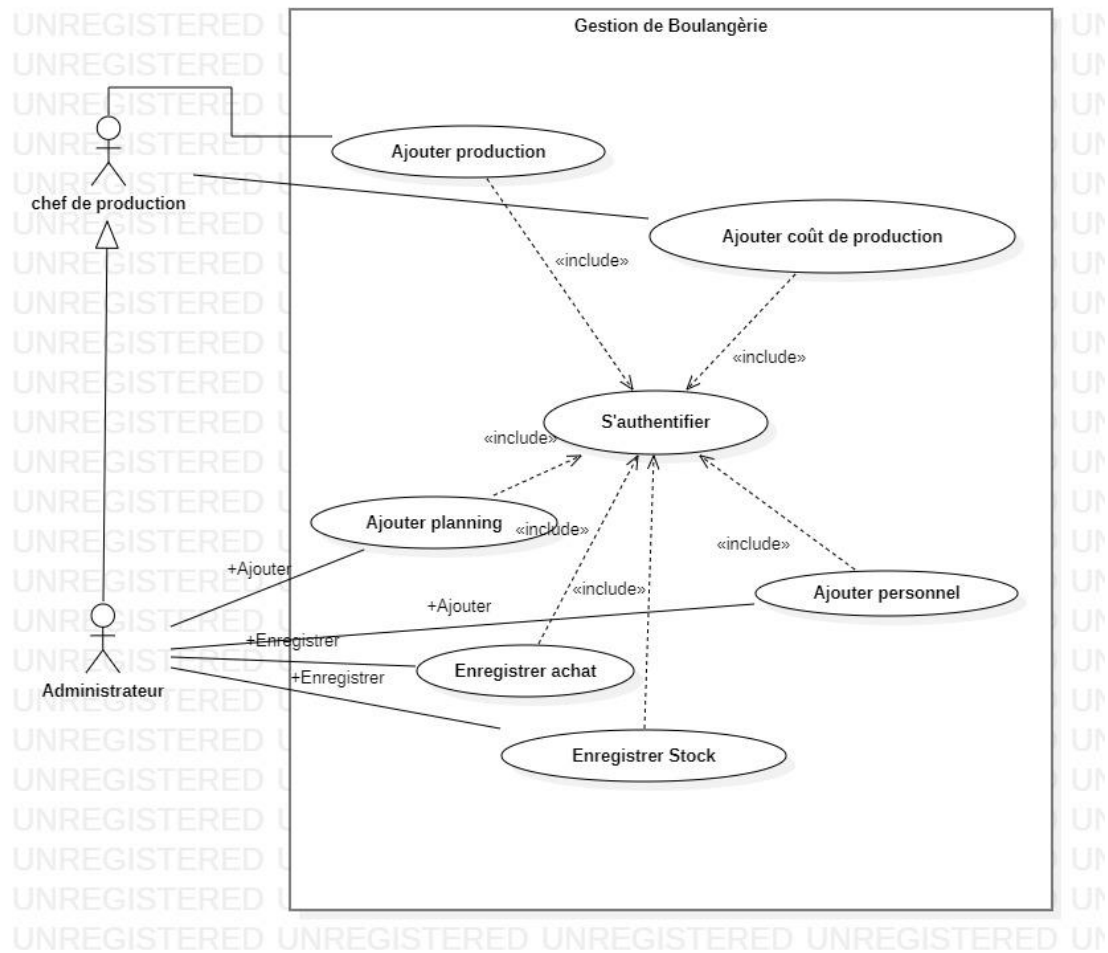


Fig 1

Page 10 of 10

Choix de la technologie et création de la base de donnée

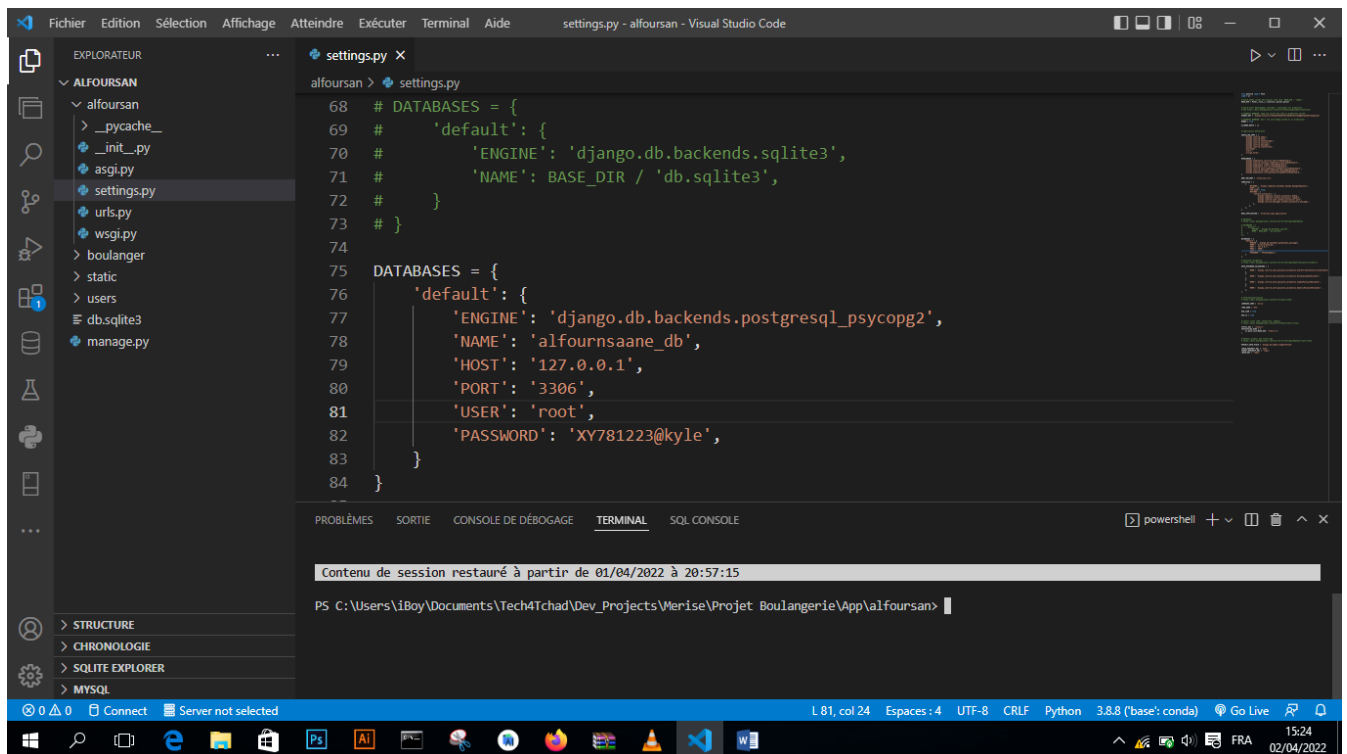
❖ Choix de la technologie

Pour ce projet, notre choix est porté sur le Framework **Django**. C'est un cadre de développement web open source, qui utilise le langage Python. Il a pour but de rendre le développement web 2.0 simple et rapide.

Il y a beaucoup des avantages de ce Framework, par exemple on peut intégrer des fonctionnalités de prédiction des chiffres d'affaire, de calcul automatisés du seuil de rentabilité à l'aide des formules mathématiques super puissantes.

❖ Création de la base de données et les tables

En Django, il y a une méthode plus facile de créer la base de donnée avec tous ces tables, il s'agit de créer la base de donnée, n'importe lequel (**MySQL, Microsoft SQL Server, MongoDB, Cassandra...**) ensuite configurer la projet Django avec la base de donnée, dans le fichier **settings.py**



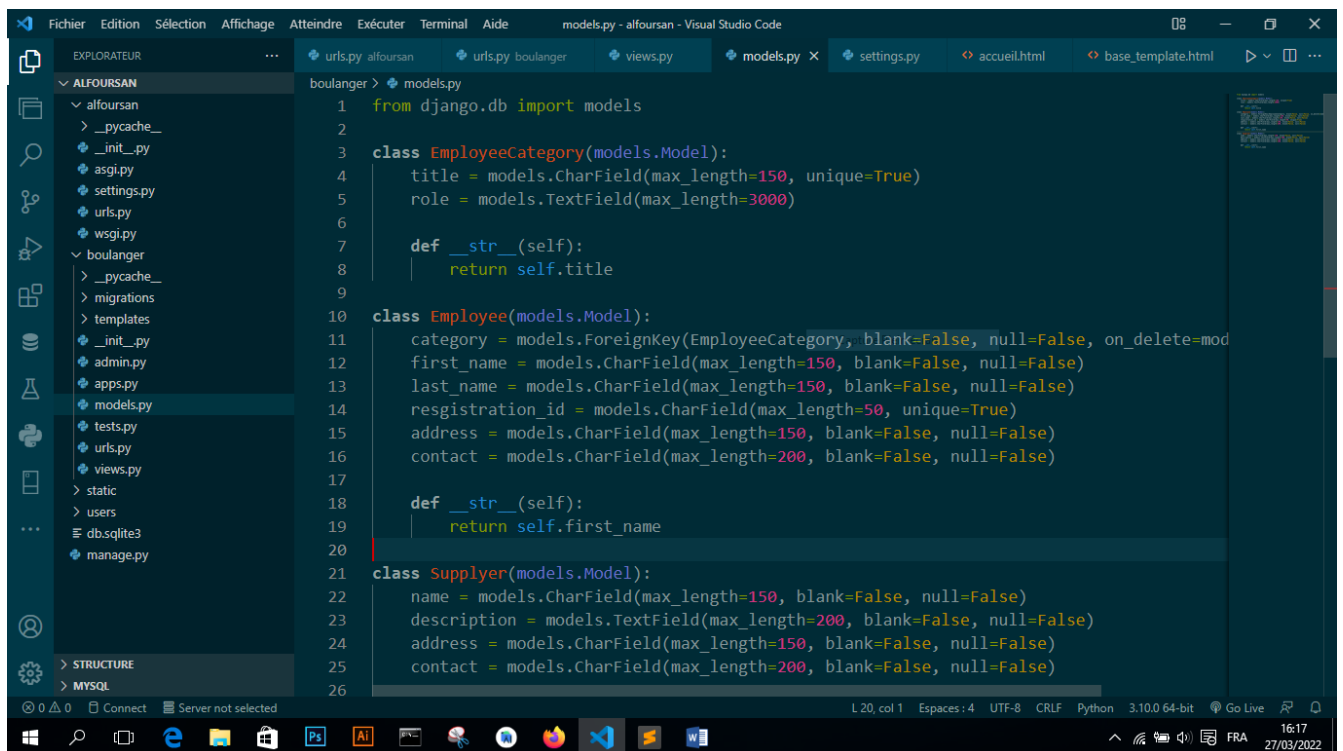
```
68 # DATABASES = {
69 #     'default': {
70 #         'ENGINE': 'django.db.backends.sqlite3',
71 #         'NAME': BASE_DIR / 'db.sqlite3',
72 #     }
73 # }
74
75 DATABASES = {
76     'default': {
77         'ENGINE': 'django.db.backends.postgresql_psycopg2',
78         'NAME': 'alfournsaane_db',
79         'HOST': '127.0.0.1',
80         'PORT': '3306',
81         'USER': 'root',
82         'PASSWORD': 'XY781223@kyle',
83     }
84 }
```

Fig 3

Configurer la partie **DATABASES** conformément avec les informations de la base de donnée.

Ensuite dans le fichier **model.py** de l'application Django, configurer tous les tables avec leurs jointures, ensuite dans la terminale exécuter cette commande **py manage.py makemigrations** pour créer la migration. Ensuite exécuter la commande **py manage.py migrate** pour créer toutes les tables de la base de donnée avec leurs jointures.

→ Exemple d'un Model Django



```
1 from django.db import models
2
3 class EmployeeCategory(models.Model):
4     title = models.CharField(max_length=150, unique=True)
5     role = models.TextField(max_length=3000)
6
7     def __str__(self):
8         return self.title
9
10 class Employee(models.Model):
11     category = models.ForeignKey(EmployeeCategory, blank=False, null=False, on_delete=models.CASCADE)
12     first_name = models.CharField(max_length=150, blank=False, null=False)
13     last_name = models.CharField(max_length=150, blank=False, null=False)
14     registration_id = models.CharField(max_length=50, unique=True)
15     address = models.CharField(max_length=150, blank=False, null=False)
16     contact = models.CharField(max_length=200, blank=False, null=False)
17
18     def __str__(self):
19         return self.first_name
20
21 class Supplier(models.Model):
22     name = models.CharField(max_length=150, blank=False, null=False)
23     description = models.TextField(max_length=200, blank=False, null=False)
24     address = models.CharField(max_length=150, blank=False, null=False)
25     contact = models.CharField(max_length=200, blank=False, null=False)
26
```

Fig 4

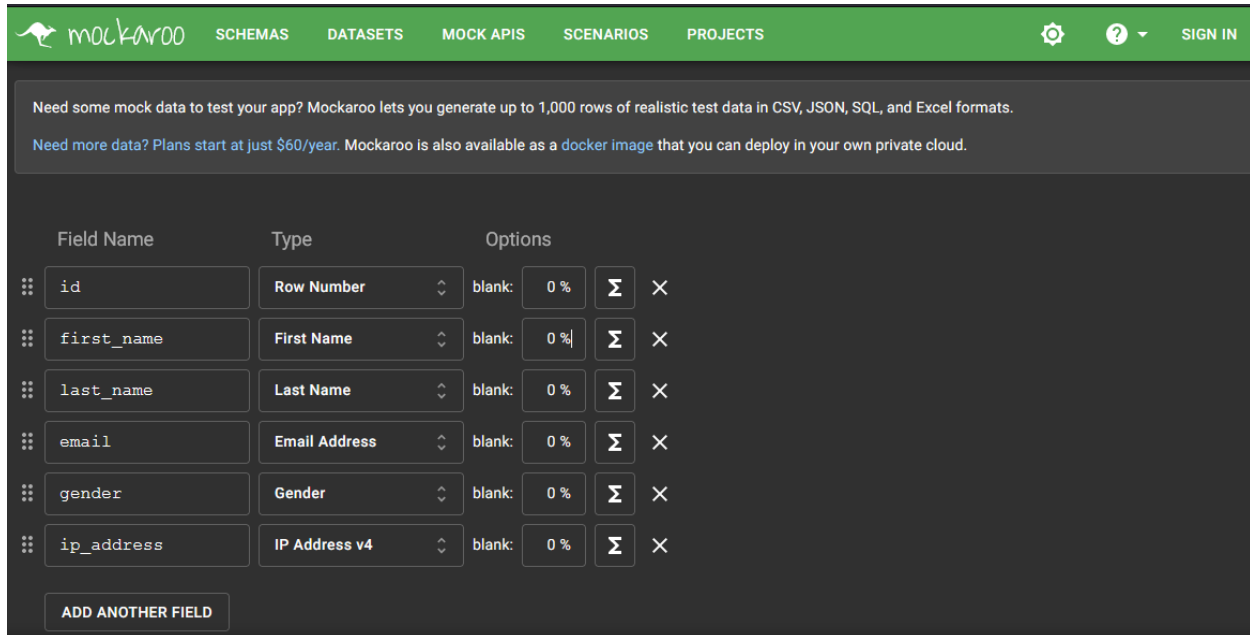
Sur cette image, les **class** sont les tables d'une base de donnée avec leurs champs. Par exemple dans la table **Employee**, on trouve le champ catégorie, prénom, nom, adresse, etc. et le champ category est une clé étrangère, comme vous pouvez le noter c'est écrit dans son paramètre **models.ForeignKey** de la table **EmployeeCategory**.

Alors supposons que nous avons utilisé PostgreSQL comme serveur de notre base de donnée, alors pour importer des données sous forme csv dans notre base de donnée, nous utiliserons PGAdmin comme interface graphique pour l'importation des données.

Insertion des données

Pour l'insertion des données, nous allons d'abord générer des données aléatoires sur le site web Mockaroo.com pour afin de peupler notre base de données.

→ Génération des données dans Mockaroo.com



The screenshot shows the Mockaroo.com web interface. At the top is a green navigation bar with the Mockaroo logo and links for SCHEMAS, DATASETS, MOCK APIS, SCENARIOS, and PROJECTS. On the right of the bar are icons for settings, help, and a 'SIGN IN' button. Below the navigation bar is a dark grey area with a light green border containing two lines of text: 'Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.' and 'Need more data? Plans start at just \$60/year. Mockaroo is also available as a docker image that you can deploy in your own private cloud.' Below this is a table configuration interface with three columns: 'Field Name', 'Type', and 'Options'. There are six rows of fields: 'id' (Row Number), 'first_name' (First Name), 'last_name' (Last Name), 'email' (Email Address), 'gender' (Gender), and 'ip_address' (IP Address v4). Each row has a 'blank:' dropdown set to '0 %', a 'Σ' icon, and an 'X' icon. At the bottom left of the table is a button labeled 'ADD ANOTHER FIELD'.

Field Name	Type	Options
id	Row Number	blank: 0 % Σ X
first_name	First Name	blank: 0 % Σ X
last_name	Last Name	blank: 0 % Σ X
email	Email Address	blank: 0 % Σ X
gender	Gender	blank: 0 % Σ X
ip_address	IP Address v4	blank: 0 % Σ X

ADD ANOTHER FIELD

Fig 5

Pour générer des données aléatoires sur Mockaroo.com, ajouter toutes les colonnes de la table sur laquelle vous allez insérer les données, en prenant en compte le type de données que la colonne accepte. Par exemple les id, généralement c'est des entiers, donc dans notre cas ici c'est le **Row Number**, ainsi de suite.

Ensuite dans le champ **# Rows:** préciser le nombre des données que vous aimeriez générer, puis le format. Le format est généralement sous CSV, mais vous avez la possibilité de générer sous d'autres formats selon le choix de votre moteur de base de données, par exemple le **JSON** pour les bases de données NoSQL, ou encore en format **SQL**, **Excel** et bien d'autres. Ensuite cliquer sur **DOWNLOAD DATA** pour générer et télécharger le fichier.

→ Insertion des données aléatoire sur notre base de données

Il y a deux méthodes pour insérer des données dans PostgreSQL, on peut le faire à l'aide de la ligne de commande, ou par importation.

Pour importer des données à partir d'un fichier CSV ou Excel, nous allons encore faire appel à pgAdmin. Sur le menu à gauche, cliquer sur **Servers**, puis **PostgreSQL**, ensuite **Databases**, la liste de toutes les bases de données créer sera afficher.

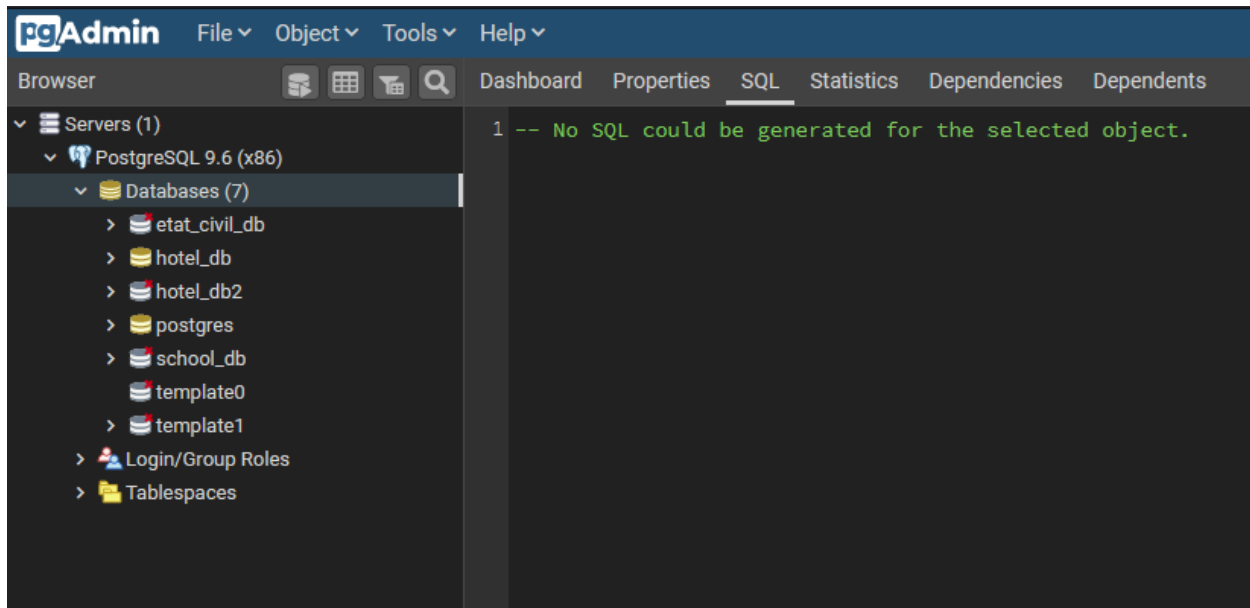


Fig 6

Cliquez sur la base de données concerné, ensuite sur **schemas**, **public**, puis **tables**. La liste de toutes les tables que vous avez créée sera affichée. Faites un clic droit sur la table dans laquelle vous voulez importer les données, ensuite cliquer sur le menu Import/Export, une fenêtre pop-up s'ouvre, charger le fichier CSV télécharger depuis le site web de Mockaroo.com, puis cliquer sur **OK** pour uploader le fichier.

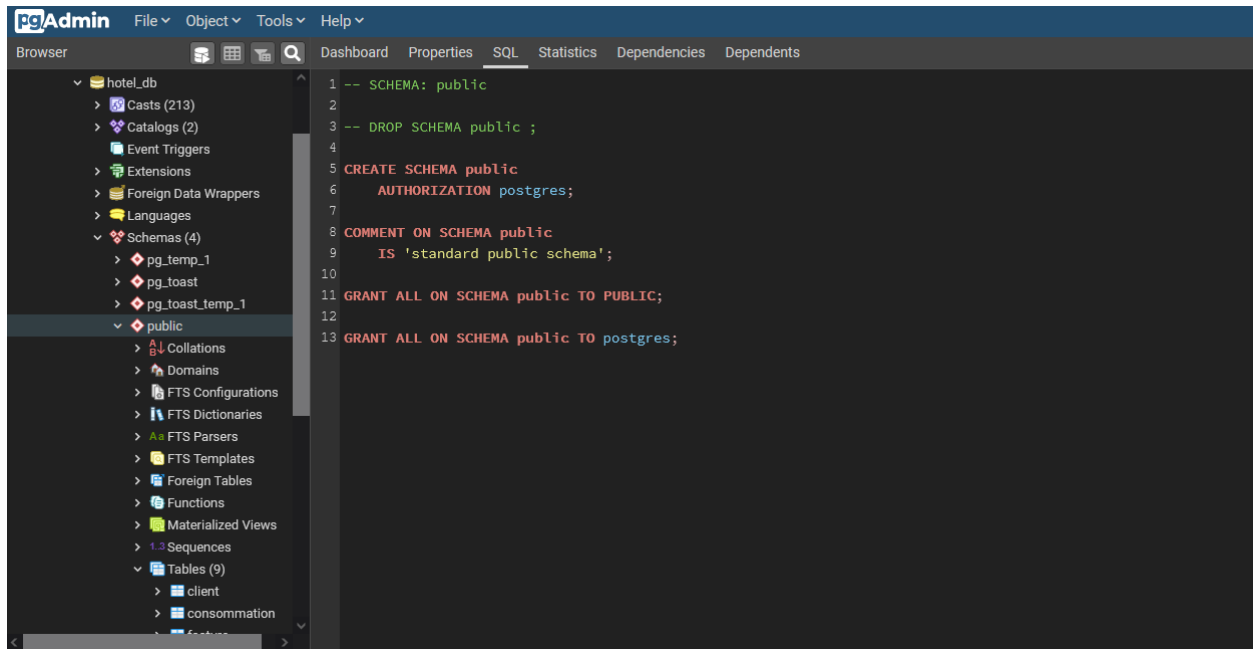


Fig 7

Difficultés rencontrées

Il n'y a pas plus difficile que de trouver une idée appropriée. La tâche était d'abord de trouver une idée de projet de conception et modélisation d'un système d'information d'une entreprise de production. Vu mes connaissances limitées dans le domaine industriel, il a fallu trouver l'idée d'une entreprise de fabrication que l'on sait à peu près son fonctionnement, pour pouvoir faire une simulation de ces activités afin de le modéliser.

Conclusion

Ce travail mené de bout en bout nous a permis de prendre un éventail de connaissances sur ce que représentent le perfectionnement et la maîtrise d'outils informatiques destinés à modéliser, créer et gérer une base de données.

De l'analyse des besoins aux applications informatiques en passant par l'élaboration d'un cahier des charges, de modélisation UML des données qui nous engageaient dans un scénario précis, nous avons pu acquérir une expérience sur le développement et la fonctionnalité de PostgreSQL, et Django Framework.

Liste des figures

- Fig 1 - capture d'écran du diagramme de cas d'utilisation.
- Fig 2 - capture d'écran du diagramme de classe.
- Fig 3 - capture d'écran du fichier settings.py du projet django.
- Fig 4 - capture d'écran du fichier models.py de l'application django.
- Fig 5 - capture d'écran du site web Mockaroo.com
- Fig 6 - capture d'écran pour l'importation des données CSV en utilisant pgAdmin
- Fig 7 - capture d'écran pour l'importation des données CSV en utilisant pgAdmin