

# OC PIZZA

## Projet 10

### Dossier d'exploitation

Version 1

**Auteur**

*Rodolphe FRANCOTTE*  
*Analyste - Programmeur*

## TABLE DES MATIÈRES

<b>OPENCLASSROOMS</b> 01.80.88.80.30 <a href="mailto:hello@notify.openclassrooms.com">hello@notify.openclassrooms.com</a>	<b>OC Pizza</b> Projet 10 <i>Dossier d'Exploitation</i>	Version : Version 1.0 Date : 05/01/2022
---	---	--

# 1 - VERSIONS

Auteur	Date	Description	Version
Rodolphe FRANCOTTE	05/01/2022	Document terminé	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application OC Pizza.

L'objectif du document est de spécifier les différentes informations nécessaires à l'exploitation en règle du système et de pouvoir réagir de manière appropriée lorsqu'un problème surgit.

### 2.2 - Références

Pour de plus amples informations, se référer :

- **DCT – OC Pizza** : Dossier de conception technique de l'application.
- **DCF – OC Pizza** : Dossier de conception fonctionnelle de l'application.
- **PV – OC Pizza** : PV de livraison de l'application.

## 3 - PRÉ-REQUIS

### 3.1 - Système

#### 3.1.1 - Serveur de Base de données

Le serveur de base de données sert à stocker, extraire et gérer les données dans la base de données. Il permet également de gérer la mise à jour des données.

**PostgreSQL\*** est le serveur de base de données qui héberge la base de données (schéma projet8\_OCPizza).

C'est un système de gestion de data base open-source. L'outil assure une excellente compatibilité avec SQL pour une requête relationnelle.

Ce système de gestion de base de données permet de gérer de nombreux types de datas avec ses fonctionnalités basiques.

Très populaire, une vaste communauté de développeurs web expérimentés a contribué au développement et à l'amélioration du système, devenant l'un des plus fiables sur le marché.

PostgreSQL fonctionne selon une architecture client/serveur, il est constitué :

- d'une partie serveur qui héberge la base de données et qui est capable de traiter les requêtes des clients. Il s'agit dans le cas de PostgreSQL d'un programme résident en mémoire appelé postmaster.
- d'une partie client qui doit être installée sur toutes les machines nécessitant d'accéder au serveur de base de données.

Les clients peuvent interroger le serveur de bases de données à l'aide de requêtes SQL.

##### 3.1.1.1 - Caractéristiques techniques

Postgresql : Version 14.1

#### 3.1.2 - Serveur Web

Un serveur Web est un serveur informatique qui héberge le contenu de l'application Web et publie celui ci en ligne sur Internet.

L'application Oc Pizza étant déployée sur Internet, il est nécessaire d'installer un tel type de logiciel.

Nous allons utiliser le serveur d'application open source Apache Tomcat pour héberger l'application web.

Il s'agit d'un serveur d'application open source basé sur un environnement Java.

Nous utiliserons également Eclipse IDE for Java EE Developers et le JDK pour nous permettre de développer l'application en Java EE.

### **3.1.2.1 - Caractéristiques techniques**

Apache Tomcat : Version 10.0.13

### **3.1.3 - Serveur de Fichiers**

Serveur Tomcat 10.0.13 hébergeant les différents fichiers de l'application web.

## **3.2 - Bases de données**

Les bases de données et schémas suivants doivent être accessibles et à jour :

- **Projet8\_OC\_PIZZA\_VF** : version 1.0
- **Nombre de table** : 13 tables
- **Interface** : PgAdmin Version 6.1

## **3.3 - Web-services**

Les web services suivants doivent être accessibles et à jour :

- **L'API Directions de Google Map** : elle fournit les itinéraires pour les livreurs et affiche les informations sur le trafic en temps réel.
- **L'API Stripe** : c'est une plateforme de traitement des paiements qui permet de transférer de l'argent du compte bancaire du client vers le compte d'une entreprise au moyen d'une transaction par carte de crédit. C'est le moyen simple d'accepter des paiements en ligne, sans frais d'installation ni frais mensuels. Stripe assure un niveau de sécurité élevé, qui permet de recevoir les paiements en toute fiabilité.

## 4 - PROCÉDURE DE DÉPLOIEMENT

### 4.1 - Déploiement de l'Application Web

#### 4.1.1 - Artefacts

L'application web est contenu dans un fichier war. Le serveur Apache Tomcat déploiera alors l'application à partir du fichier war. grâce à ses bibliothèques et son compilateur permettra de transformer le code destiné à la machine virtuelle Java. Les données de la base seront lues par l'intermédiaire du driver JDBC permettant au serveur Apache Tomcat d'y avoir accès.

#### 4.1.2 - Environnement de l'application web

##### 4.1.2.1 - JDK (Java Development Kit)

Installer la version 8 du JDK. Pour cela, se rendre sur [le site d'Oracle](#) et télécharger la version 8.

Nous nous assurons que la variable d'environnement Java est correctement configurée :

1. Localiser notre répertoire d'installation de Java et copiez le chemin d'accès. Le chemin par défaut est C:\Program Files\Java\jdk <numéro de version>.
2. Depuis le menu Démarrer de Windows, accéder à Paramètres système avancés. La manière d'accéder à ces propriétés varie en fonction de votre version de Windows. Par exemple, sous Windows 10, recherchez Variables d'environnement, puis sélectionnez Modifier les variables d'environnement système.
3. Cliquons sur Variables d'environnement.
4. Dans la section Variables système, cliquons sur Nouvelle.
5. Dans le champ Nom de la variable, spécifions JAVA\_HOME.
6. Dans le champ Valeur de la variable, spécifions le chemin d'accès à notre installation Java.
7. Cliquez sur OK.

##### 4.1.2.2 - Serveur Apache Tomcat

Installer ApacheTomcat sur votre système. Pour cela, télécharger la version souhaitée [dans ce lien](#). Ensuite suivre le protocole d'installation, [à partir de ce lien](#).

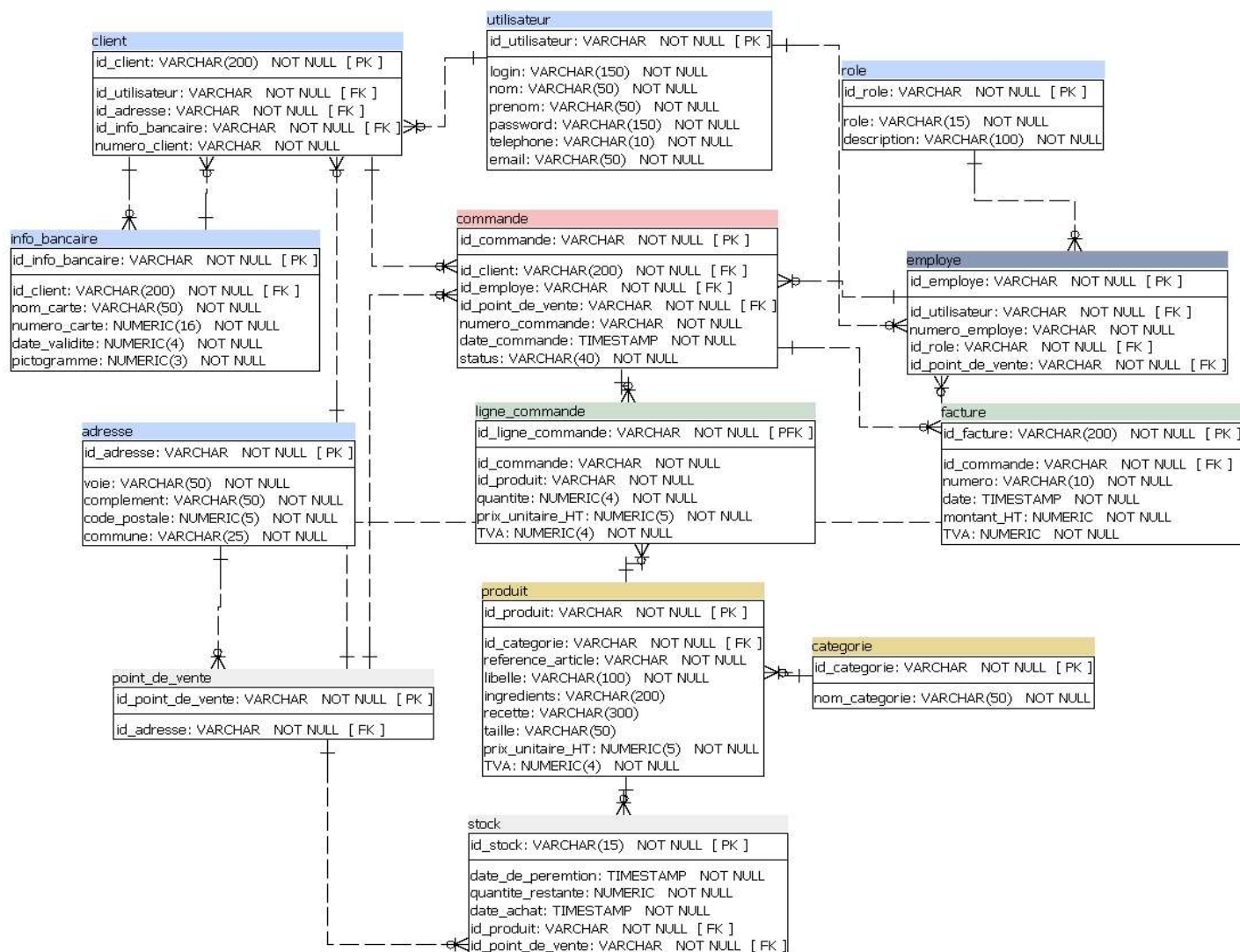
Une fois le serveur installé, il faut permettre au serveur d'accéder à la base de données, à l'intérieur du dossier Tomcat, ouvrons le dossier lib puis ajoutons le fichier postgresql-42.2.1.jar que nous trouvons [à cette adresse](#) dans la colonne intitulée 'JDBC 42'.

### 4.2.2.3 - Base de données

#### Power Architect

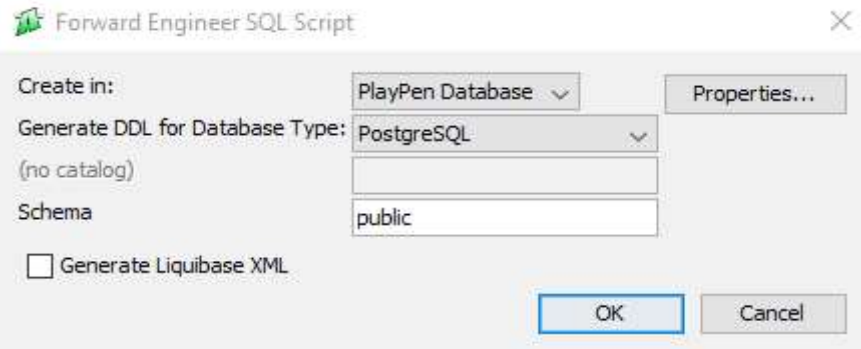
A partir de notre modèle physique de données, créer la structure de la base de données.

Pour cela, reproduire dans le logiciel 'Power Architect' que nous [téléchargeons ici](#), le modèle physique de données.



Ensuite, cliquer sur l'icône 'SQL' et remplir les champs, comme indiqué sur l'image ci-dessous :





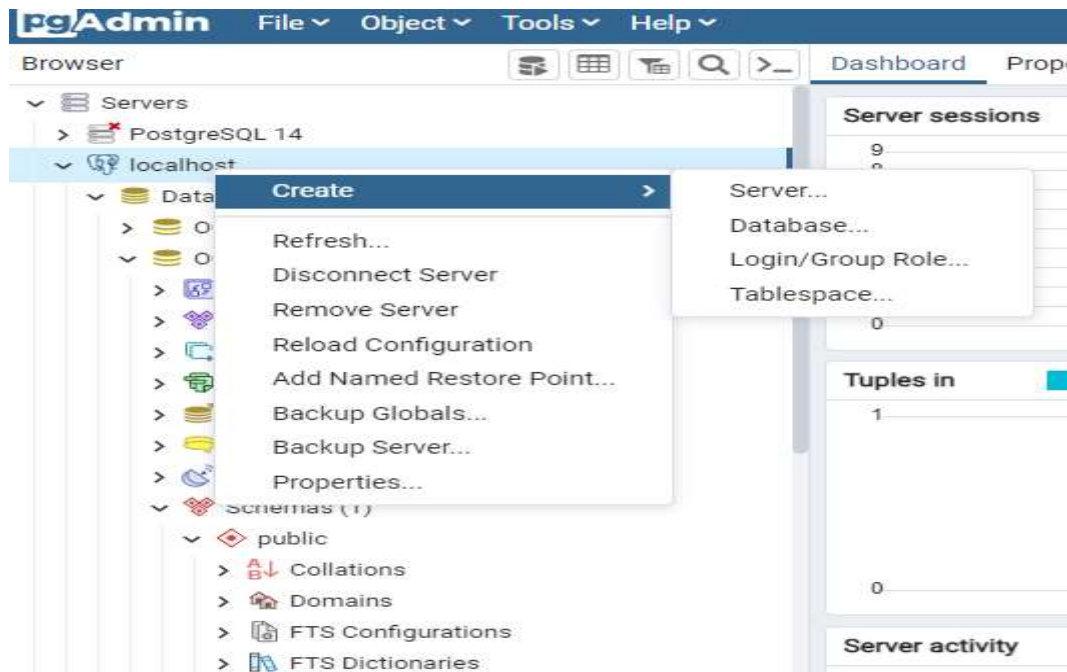
Power Architect va ensuite générer la requête Postgresql permettant la création de la structure de la base de données, faire une copie entière de la requête.

### Création Base de données

1 - Télécharger tout d'abord les logiciels nécessaires. Pour cela se rendre à cette [adresse](#), puis sélectionner notre système et la version 4 de pgAdmin.

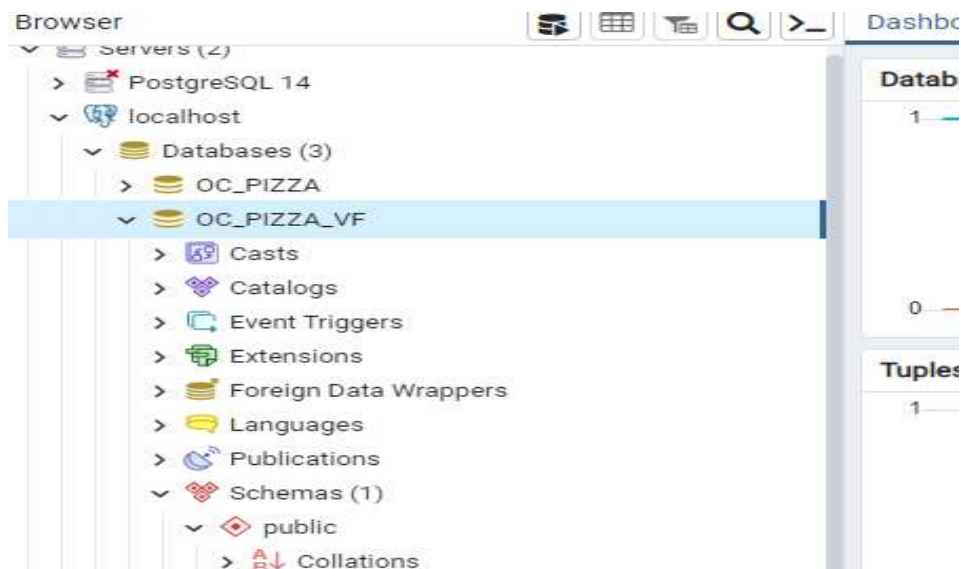
Le programme installe l'interface pgAdmin et le système de gestion de base de données Postgresql.

2 – Se rendre sur l'interface de pgAdmin puis créer la base de données.



Il ne reste plus qu'à lui donner un nom et valider.

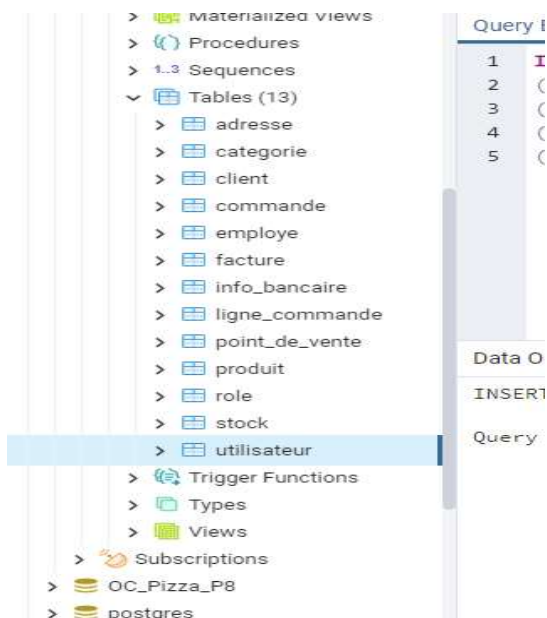
3 - Maintenant que la base est créée, dans la colonne de gauche, sélectionner notre base de données et cliquer sur l'icône en haut à droite de l'image ci-dessous :



Ensuite, coller la requête générée par Power Architect.

Sélectionner l'ensemble de la requête et exécuter.

La structure est terminée, toutes les tables composant la base de données sont disponibles.



#### 4 – Ajouter les datas dans la base de données.

Coller la requête dans l'interface pgAdmin, sélectionner le tout, puis exécuter. Toutes les données de la base de données sont désormais accessibles.

Data Output		Explain	Messages	Notifications	
	<div><div><div><div></div><div></div></div><div><div>id</div><div>[PK] numeric</div></div></div><div></div></div>	<div><div><div><div></div><div></div></div><div><div>voie</div><div>character varying (70)</div></div></div><div></div></div>	<div><div><div><div></div><div></div></div><div><div>complement</div><div>character varying (50)</div></div></div><div></div></div>	<div><div><div><div></div><div></div></div><div><div>code_postal</div><div>numeric (5)</div></div></div><div></div></div>	<div><div><div><div></div><div></div></div><div><div>commune</div><div>character varying (25)</div></div></div><div></div></div>
1		116 avenue du val de saone	batiment C	69580	Sathonay camp
2		216 rue de la montée	1 étage	69580	Sathonay Village
3		346 rue de la république	bat C	69300	Caluire
4		456 rue de la montée	4 étages	69300	caluire
5		566 rue de la Ligote	8 étages	69500	Bron
6		61 impasse de la vie	maison au fond de l'impasse	69330	Meyzieu
7		718 square marechal Lyautey	bat A	69300	Caluire
8		816 rue de la République	OC Pizza de Lyon Terreau	69001	Lyon
9		9164 grande rue de la Croix-R...	OC Pizza de Lyon Croix-Rous...	69002	Lyon
10		1010 rue Victor Hugo	OC Pizza de Lyon Bellecour	69004	Lyon
11		1125 cours Charlemagne	OC Pizza de Lyon Perrache	69002	Lyon
12		12CC de Lyon Part-Dieu	OC Pizza de Lyon Part-Dieu	69003	Lyon

## 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

### 5.1 - Base de données

Une fois les identifiants et l'adresse saisis dans les fichiers `application.properties`, l'accès à la base de données est automatique.

L'accès à la base de données se fait par l'intermédiaire de l'interface pgAdmin 4, en saisissant les informations d'authentification demandées.

### 5.2 - Application web

L'application web doit être démarrée ou arrêtée par l'intermédiaire du serveur complet.

## 6 - PROCÉDURE DE MISE À JOUR

### 6.1 - Base de données

La mise à jour de la version de la base de données est mise à jour par le biais du fichier pom.xml.

Il suffit de modifier la version de la dépendance avec la version souhaitée dans le fichier pom.

### 6.2 - Application web

L'application web est mise à jour par l'intermédiaire des différents modules.

## 7 - SUPERVISION/MONITORING

### 7.1 - Supervision de l'application web

Afin de tester que l'application web est toujours fonctionnelle, exécuter à partir de notre IDE les tests automatisés et s'assurer qu'ils soient tous vert.

Sinon, à partir d'une console, entrer le chemin du dossier, puis saisir la ligne de commande : `mvn test`

S'assurer que le résultat affiché ne comporte aucune erreur.

## 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

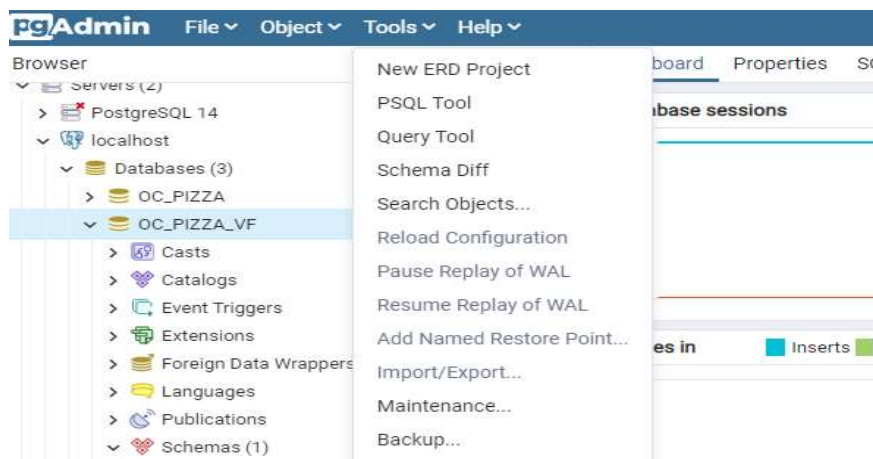
### 8.1 - GitHub

L'ensemble du projet est sauvegardé sur GitHub. Vous pouvez le restaurer à tout moment.

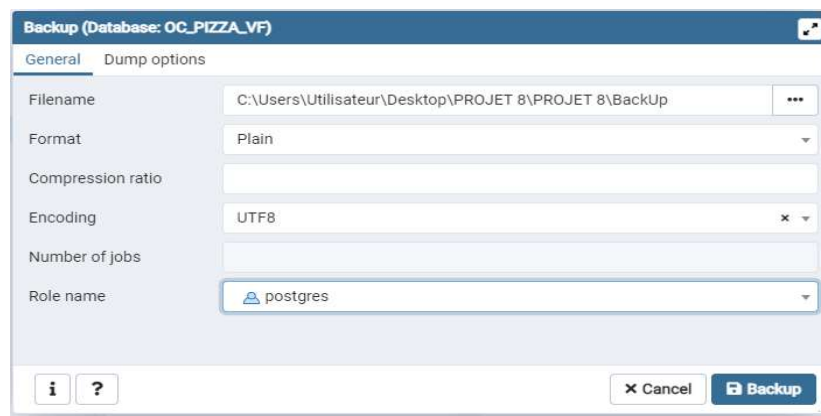
### 8.2 - Base de données

Nous pouvons également sauvegarder les modifications que nous allons apporter à la base de données, suivant la procédure suivante :

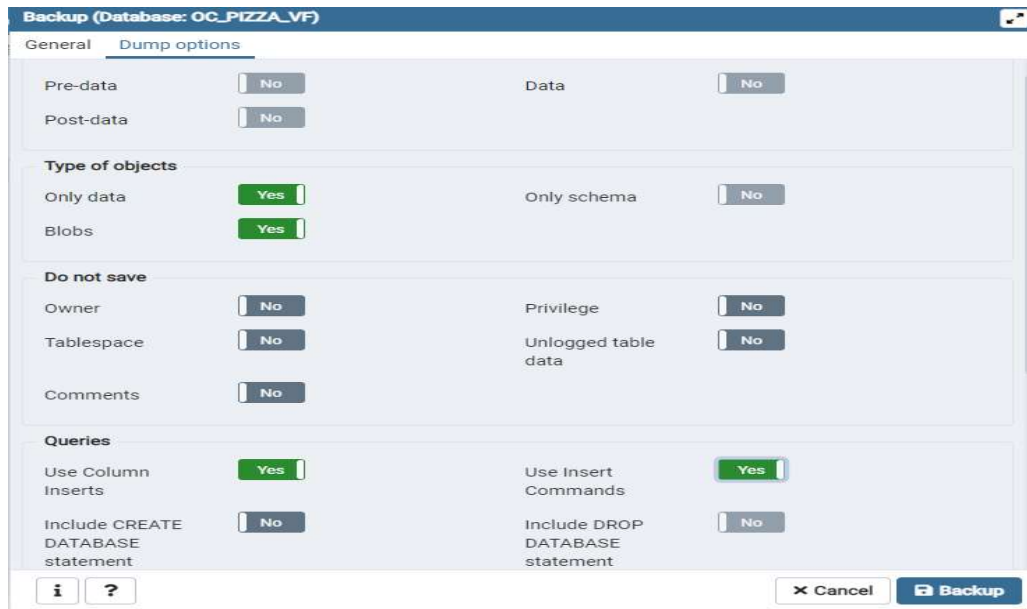
- Par l'intermédiaire de pgAdmin 4, aller dans l'onglet 'Tools' et sélectionner 'Backup'.



- Ensuite remplir les différents champs du premier onglet, le champ 'Filename' est le chemin de votre sauvegarde plus le nom de votre fichier, pour le reste remplir selon l'exemple ci-dessous :



- Dans le deuxième onglet, modifier les boutons coulissant comme sur l'exemple ci-dessous :



**Backup (Database: OC\_PIZZA\_VF)**

General | **Dump options**

Pre-data	<input type="button" value="No"/>	Data	<input type="button" value="No"/>
Post-data	<input type="button" value="No"/>		
<b>Type of objects</b>			
Only data	<input checked="" type="button" value="Yes"/>	Only schema	<input type="button" value="No"/>
Blobs	<input checked="" type="button" value="Yes"/>		
<b>Do not save</b>			
Owner	<input type="button" value="No"/>	Privilege	<input type="button" value="No"/>
Tablespace	<input type="button" value="No"/>	Unlogged table data	<input type="button" value="No"/>
Comments	<input type="button" value="No"/>		
<b>Queries</b>			
Use Column Inserts	<input checked="" type="button" value="Yes"/>	Use Insert Commands	<input checked="" type="button" value="Yes"/>
Include CREATE DATABASE statement	<input type="button" value="No"/>	Include DROP DATABASE statement	<input type="button" value="No"/>

Vérifier dans le dossier correspondant au chemin que nous avons entré si le fichier de sauvegarde a bien été créé.



## 9 - GLOSSAIRE

<b>API</b>	API (Application Programming Interface) Interface permettant à un logiciel d'accéder aux services d'un autre logiciel par le biais d'un ensemble normalisé de classes, de méthodes, de fonctions et de constantes.
<b>GitHub</b>	GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.
<b>JDBC</b>	JDBC (Java DataBase Connectivity) est une API Java constituée d'un ensemble d'interfaces et de classes permettant de se connecter à n'importe quelle base de données et d'interagir avec elle.
<b>Postgresql</b>	PostgreSQL est un système de gestion de base de données relationnelle et objet permettant la gestion de la base de données.
<b>Power Architect</b>	Logiciel qui permet de modéliser une base de données puis de générer automatiquement le schéma dans le système de gestion de base de données de notre choix.
<b>Tomcat</b>	Serveur d'application permettant d'exécuter des applications Web développées avec les technologies Java.