

Projet réseau PEI-2 2022-2023

PROTOCOLES RÉSEAUX IP, SQL & HTML

1. OBJECTIFS DU PROJET

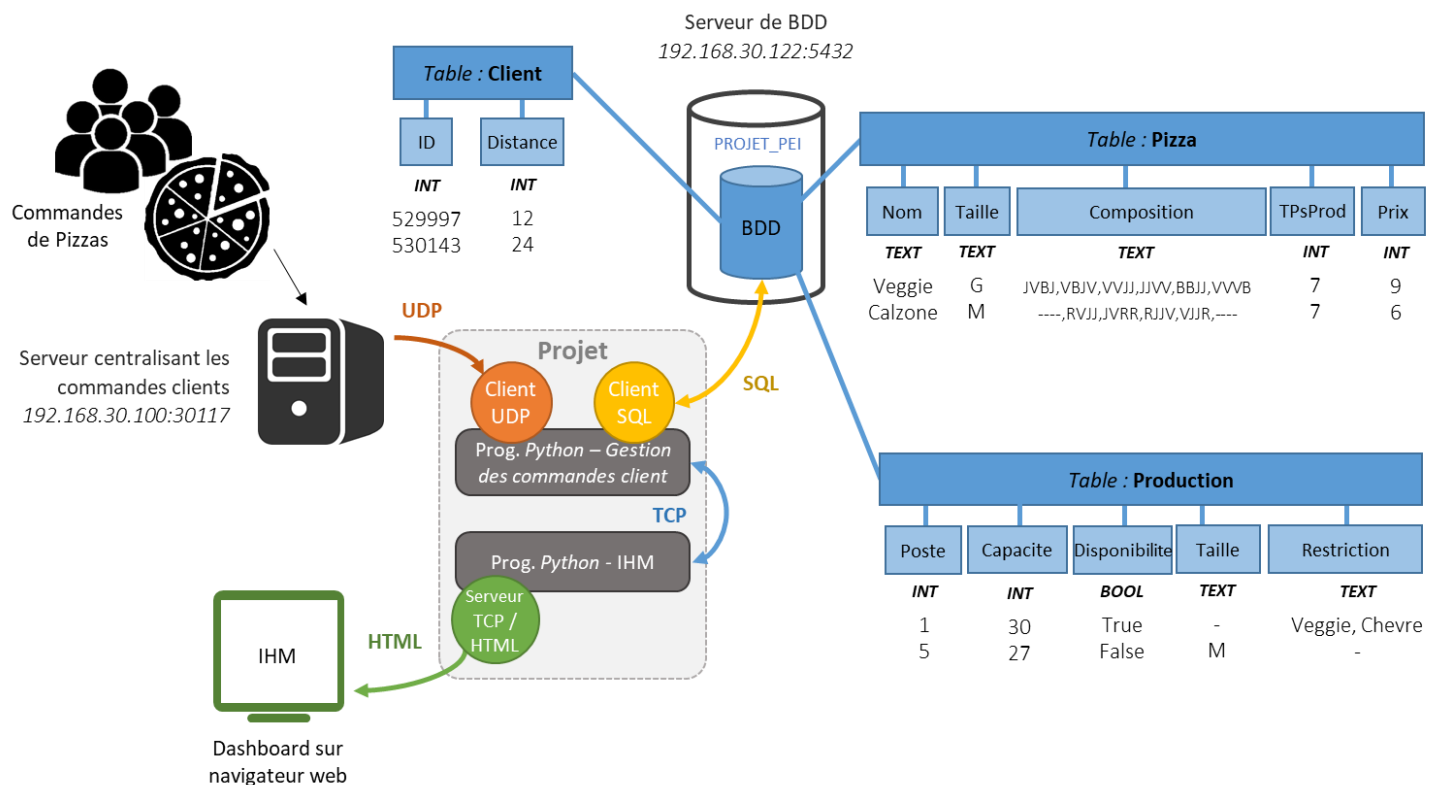
Ce projet a pour objectif de vous faire manipuler des notions de réseaux et de programmation que vous avez abordées en TP. Vous serez amenés à récupérer des informations depuis un serveur UDP (*Client/Serveur Python*), à traiter et analyser les données reçues en les comparant à des données stockées en bases de données (*SQL*) pour enfin afficher des informations dans une IHM (*HTML*).

2. DESCRIPTION DU PROJET

Pour ce projet, vous allez être au cœur de la gestion de commandes de pizzas. Votre objectif est de valider les commandes clients ; en fonction de la disponibilité des moyens de productions et du délai de livraison. Vous générerez par la suite des statistiques de ventes et des indicateurs de performance de la production.

Pour cela, un serveur UDP reçoit et concentre les commandes de pizzas de clients. Les informations concernant les utilisateurs, les pizzas et les capacités des moyens de production sont stockées dans une base de données SQL.

Votre travail sera dans un premier temps, de valider la faisabilité des commandes clients en fonction de l'heure souhaitée de la livraison et du temps de livraison. Dans un second temps, vous gèrerez les capacités et les disponibilités des moyens de production. Enfin, vous afficherez des informations et statistiques de ventes sur une IHM.



DÉROULÉ DU PROJET

Le Projet Réseau d'une durée de 20h se déroulera sur 5 séances encadrées successivement par M. EVENO I. (iehann.eveno@univ-ubs.fr) et par M. TOUBLANC T. (thomas.toublanc@univ-ubs.fr).

Attention : ceci n'est pas un « Méga-TP », il s'agit bien d'un projet ; c'est-à-dire que votre notation prendra en compte votre gestion de projet autant que vos rendus. Ceux-ci devront comprendre un rapport détaillant votre méthodologie, les outils utilisés, les problématiques rencontrées et l'explication de vos codes.

Description détaillée du projet :

I – Récupérer les informations du serveur UDP

Un serveur UDP (192.168.30.100:30117) concentre toutes les commandes de pizzas des clients sous le format « *Date Heure,ID_Client,Nom_Pizza,Taille,Quantité,Heure_Livraison_Souhaitée* ». Exemple :

26/11/2022 10:03:12,530080,Orientale,G,4,10:40

Votre premier objectif est de récupérer les informations disponibles sur le serveur en rédigeant le code Python d'un client UDP. Vous aurez alors en temps-réel les commandes de pizzas de près de 300 clients.

Pour votre programme (*ClientUDP.py*), vous devrez notamment utiliser la fonction *split()* pour générer vos variables *Date*, *Heure*, *ID_Client*, *Nom_Pizza*, *Taille*, *Quantité* et *Heure_Livraison_Souhaitée*.

II – Premiers pas avec une Base de Données

Prenez en main le langage SQL en étudiant le programme *CodePython_BDD_PEI.py* que vous trouverez dans le Moodle du Projet. L'étude et vos commentaires sur ce programme rentrera dans la notation de votre projet.

III – Dialoguer avec la Base de Données

Rédigez un second programme (*ClientSQL.py*) qui vous permette d'accéder à la table *Client* de la BDD *PROJET_PEI* (192.168.30.122:5432) et de lire le temps de livraison (champ *Distance* – en minute) des clients. La table *Client* est sous le format ;

Champs :	ID	Distance
Type :	INT	INT
Valeur (ex) :	529996	24
Valeur (ex) :	530111	12

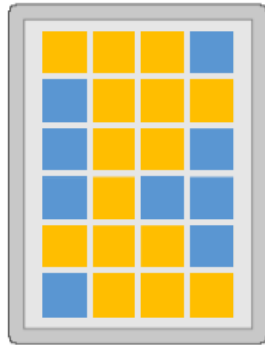
Récupérez par la suite le temps de production des pizzas commandées par les clients (champ *TPsProd* – en minute) en accédant à la table *Pizza*. Cette table est sous le format ;

Champs :	Nom	Taille	Composition	TPsProd	Prix
Type :	TEXT	TEXT	TEXT	INT	INT
Valeur (ex) :	4_Fromages	G	JJJB,BJJJ,BJJB,BJBB,JJJJ,BJJJ	9	12
Valeur (ex) :	Chevre	M	----,BJJJ,JJVV,BVJJ,JBBJ,---	7	8

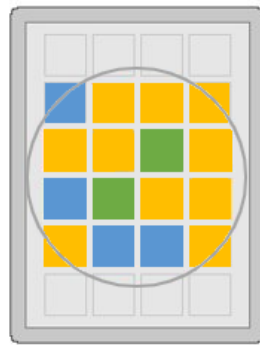
La table *Pizza* comporte des informations sur ; le prix en euro des pizzas, leurs compositions et leurs temps de production. Le tout en fonction des tailles des pizzas ; *G* pour « grande » et *M* pour « moyenne ».

Pour les deux pizzas présentées ci-dessus (4_Fromages et Chevre), leur composition peut s'illustrer selon la représentation ci-après. Pour cela, le champ *Composition* représente *en ligne* les ingrédients utilisés dans les pizzas ;

- 'J' pour *jaune* peut correspondre à de l'emmental ou des tranches d'ananas
- 'V' pour des poivrons
- etc



Représentation de la composition
de la pizza 4_Fromages (taille G)



Représentation de la composition
de la pizza Chevre (taille M)

La finalité du programme *ClientSQL.py* sera de créer une fonction mixant vos deux requêtes SQL vous permettant de valider, pour une heure de livraison donnée, la faisabilité d'une livraison ; selon le temps de production de la pizza (en prenant en compte sa taille) ainsi que du temps de livraison au client.

Faisabilité = Votre_fonction(ID_Client, Nom_Pizza, Quantite, Taille_Pizza, Heure_Livraison_Souhaitée)

IV – Concaténation des programmes « Clients »

Concaténez vos deux programmes clients (UDP et SQL) pour commencer à réaliser le programme du projet qui ; lorsque le serveur UDP envoie une nouvelle commande, vous puissiez définir si celle-ci est "acceptée" ou "non-acceptée". Réalisez alors une IHM textuelle simple sur le Terminal Linux.

V – Capacités de production et choix du poste de production

L'usine qui produit les pizzas possède six postes de production. Des informations sur ces postes sont stockées dans la table *Production*.

Sur ces postes, il est possible de fabriquer plusieurs pizzas en même temps, c'est ce que l'on nomme « Capacité ». Un poste peut fabriquer une ou plusieurs tailles de pizza (Champ *Taille* = *M* ou *G*, ou les deux : « - »). Pour causes d'allergies, certains postes ne peuvent produire certaines pizzas ; le champ *Restriction* indique alors les pizzas non-fabricables par le poste (« - » si pas de restriction particulière). Enfin, le champ *Disponibilite* informe si le poste est opérationnel (=TRUE) ou non.

Champs :	Poste	Capacite	Disponibilite	Taille	Restriction
Type :	INT	TEXT	BOOL	TEXT	TEXT
Valeur (ex) :	3	18	TRUE	G	Chevre,4_Fromages
Valeur (ex) :	4	15	FALSE	-	-

Créez des fonctions vous permettant d'attribuer les commandes clients aux bons postes de productions. Vous prendrez en compte les restrictions des postes ainsi que leur capacité maximale et leur disponibilité.

Aidez-vous des *listes* sous Python afin d'allouer les commandes de pizzas aux postes.

Exemple d'utilisation des fonctions *append()* et *remove()* de la classe *list* ;

```
My_list = []
My_list.append(['Abc',12])
My_list.append([17,'D', 'E'])
My_list.remove(['Abc',12])
print(My_list)
print(My_list[0][1])
```

Par poste, utilisez des *listes* pour stocker (*au minima*) ; la quantité ainsi que l'heure de fin de production (*heure de la commande + temps de production de la pizza*) des commandes attribuées au poste.

Supprimez la commande de votre liste lorsque l'heure de fin de production est passée. Utilisez la bibliothèque *datetime* de Python pour gérer des événements sur des horaires.

Mettez à jour votre fonction *Faisabilité* afin de tester si au moins un poste peut prendre la commande client.

VI – Statistiques et IHM (*Interface Humain-Machine*)

En adaptant le code *ExServeurTCP_HTML.py* disponible sur le Moodle, développez en *html* un *dashboard* simple, affichant le nombre total de commandes acceptées et refusées. Pour ce faire, vous devrez faire communiquer vos deux programmes (programme *ClientsUDP_SQL* et le programme de l'IHM) à travers le mécanisme de *Client/Serveur TCP* vu en TP.

Codes Optionnels

Pour ce projet, vous devrez réaliser au minimum l'une des options décrites ci-dessous. Vous avez le choix de réaliser celle qui vous semble la plus pertinente selon les compétences que vous souhaitez développer.

(Option 1) – Statistiques avancées & Amélioration de l'IHM

Réalisez des statistiques sur la gestion des commandes (Nombre d'ingrédients utilisés par poste, Total des ventes, Quantité produite, etc.) et intégrez ces informations à votre IHM.

(Option 2) – Gestion de listes d'attente

Améliorez la gestion des commandes lorsque la capacité des postes est dépassée, en mettant en attente les commandes qui pourraient être réalisables plus tard ; c.à.d. lorsque la capacité des postes le permettrait.

(Option 3) – Optimiser la commande des clients

Certains clients commandent plusieurs pizzas. Celles-ci peuvent être produites sur différents postes à différents moments. Assurez-vous que le timing de production permette au client de recevoir toutes ses pizzas bien chaudes.