

# PROJET GRENOBLE EATS

**INP**  
Ensimag

Grenoble  
**Eats**

# Quelques contraintes

## Sur les données :

### Toutes les heures et dates :

Int qui représente le temps écoulé depuis une date et heure de référence.

### Note restaurant :

La moyenne des notes des évaluations de ce restaurant.

### Tous les id :

Des int unique.

## Sur les relations :

### Type commande cohérente avec le resto :

Le type d'une commande d'un restaurant doit être dans les types possible du restaurant

### Suppression client :

On supprime le client de nos base de données mais on garde ses évaluations en les associant à un nouvel id client créée pour l'occasion.

### Pas d'évaluation pour les commandes annulés :

Si le statut de la commande est annulé alors pas d'évaluation

# Dépendances fonctionnelles :

Id resto → nom resto, tel resto, adresse resto, nombre de places, notes resto  
->> type commande  
->> évaluations resto  
->> catégorie  
->> plages horaires(plus flexible, peut tout décrire)  
->> id plat

id client → mdp, nom client, prénom client, mail client

# Dépendances fonctionnelles :

(id commande, id restaurant)

→ date commande, heure commande, heure terminée, client commande,  
type commande, prix commande, statut, adresse livraison

->> contenu(id plats)

Id plat → nom plat, description plat

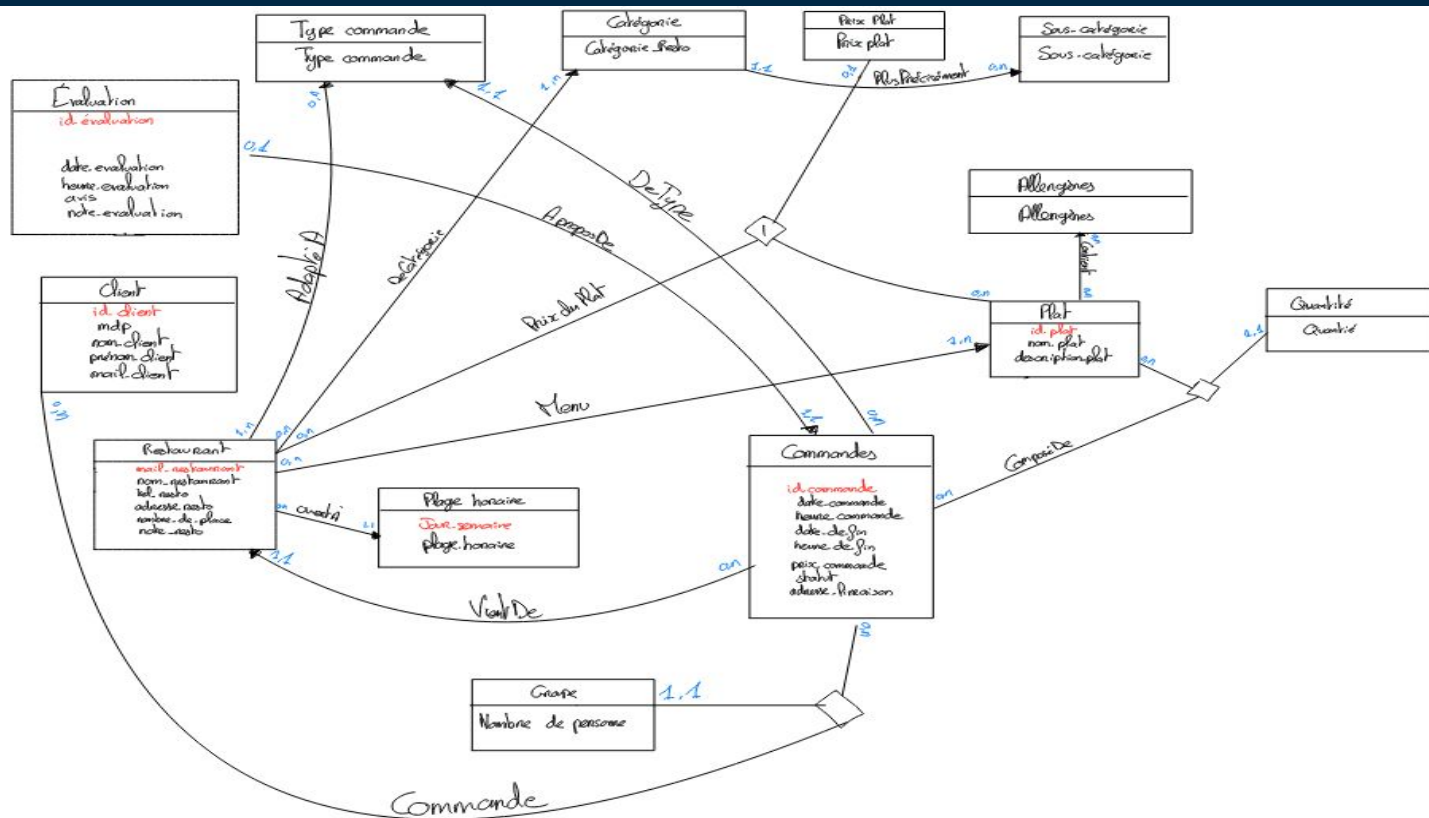
->> liste allergènes

->> ( id restaurant, prix plat) //car même plat mais prix différents  
dans différents restos

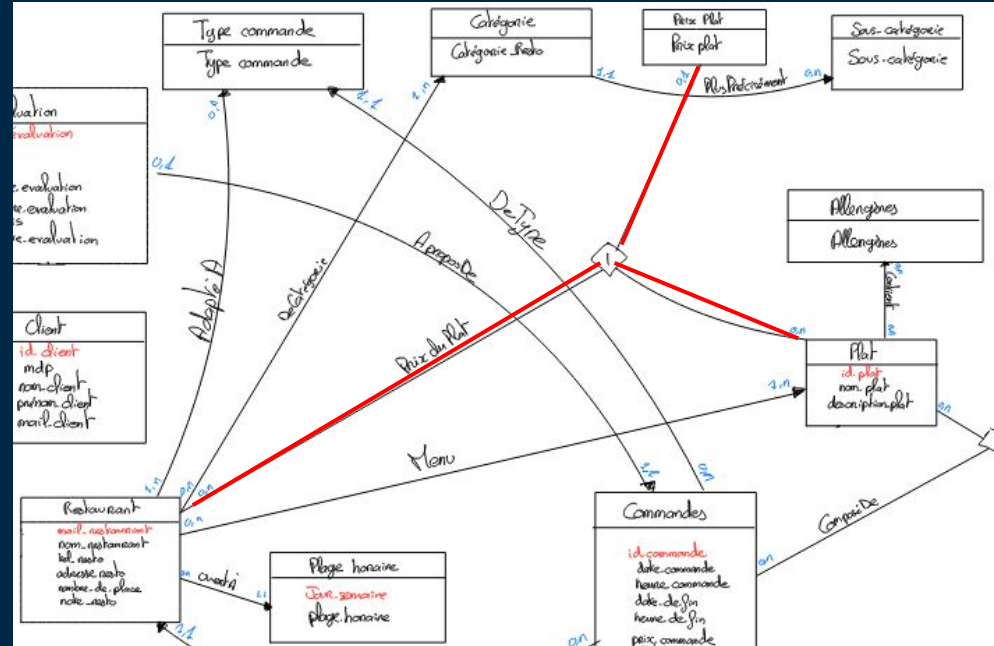
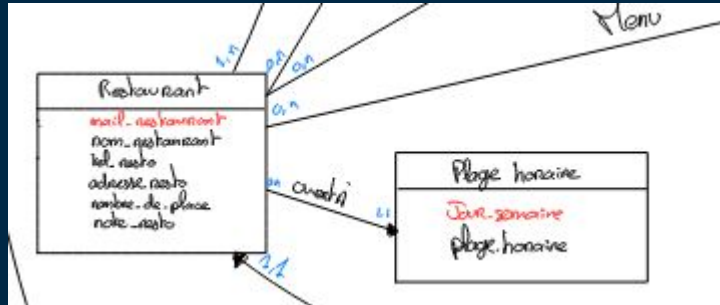
# Dépendances fonctionnelles :

(id évaluation, id restaurant)  $\rightarrow$  id commande, date évaluation, heure évaluation, avis, note évaluation

# Diagramme entité-relation



# Les points importants



# Les Tables

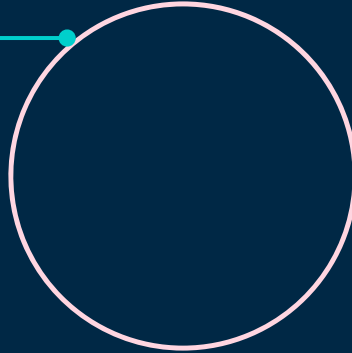
Restaurant	Evaluation	ComposéDe	Menu	OuvertA	CatégorieResto	Client
<i>mail_resto</i>	<i>id_evaluation</i>	<i>id_commande</i>	<i>mail_resto</i>	<i>mail_resto</i>	<i>mail_resto</i>	<i>id_client</i>
nom_resto	date_evaluation	id_plat	<i>id_plat</i>	<i>jour_semaine</i>	categorie_fille	mdp
tel_resto	avis	Quantité	nom_plat	horraire_midi_debut		nom_client
adresse_resto	note_evaluation		prix_plat	horraire_midi_fin		prenom_client
nbr_place				horraire_soir_debut		
note_resto				horraire_soir_fin		
CommandeClient	Commande	Plat	Allergenes	TypeCommandePossible	RelationCategorie	
<i>id_client</i>	<i>id_commande</i>	<i>id_plat</i>	<i>id_plat</i>	<i>mail_resto</i>	<i>Categorie_fille</i>	
<i>id_commande</i>	date_commande	nom_plat	allergenes	type_commande	Categorie_mere	
nbr_personnes	date_fin	description_plat				
	prix_commande					
	adresse_livraison					
	mail_resto					



# Formes normales

1FN

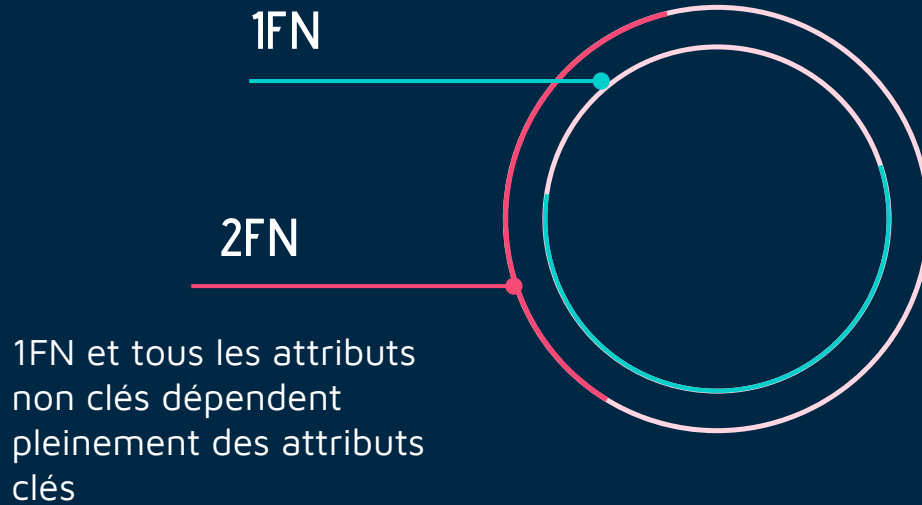
Tous les attributs ont des  
valeurs atomiques et non  
nulles



# 1FN

Restaurant	Evaluation	ComposéDe	Menu	OuvertA	CatégorieResto	Client
mail_resto	id_evaluation	id_commande	mail_resto	mail_resto	mail_resto	id_client
nom_resto	date_evaluation	id_plat	id_plat	jour_semaine	categorie_fille	mdp
tel_resto	avis	Quantité	nom_plat	horraire_midi_debut		nom_client
adresse_resto	note_evaluation		prix_plat	horraire_midi_fin		prenom_client
nbr_place				horraire_soir_debut		
note_resto				horraire_soir_fin		
CommandeClient	Commande	Plat	Allergenes	TypeCommandePossible	RelationCategorie	
id_client	id_commande	id_plat	id_plat	mail_resto	Categorie_fille	
id_commande	date_commande	nom_plat	allergenes	type_commande	Categorie_mere	
nbr_personnes	date_fin	description_plat				
	prix_commande					
	adresse_livraison					
	mail_resto					

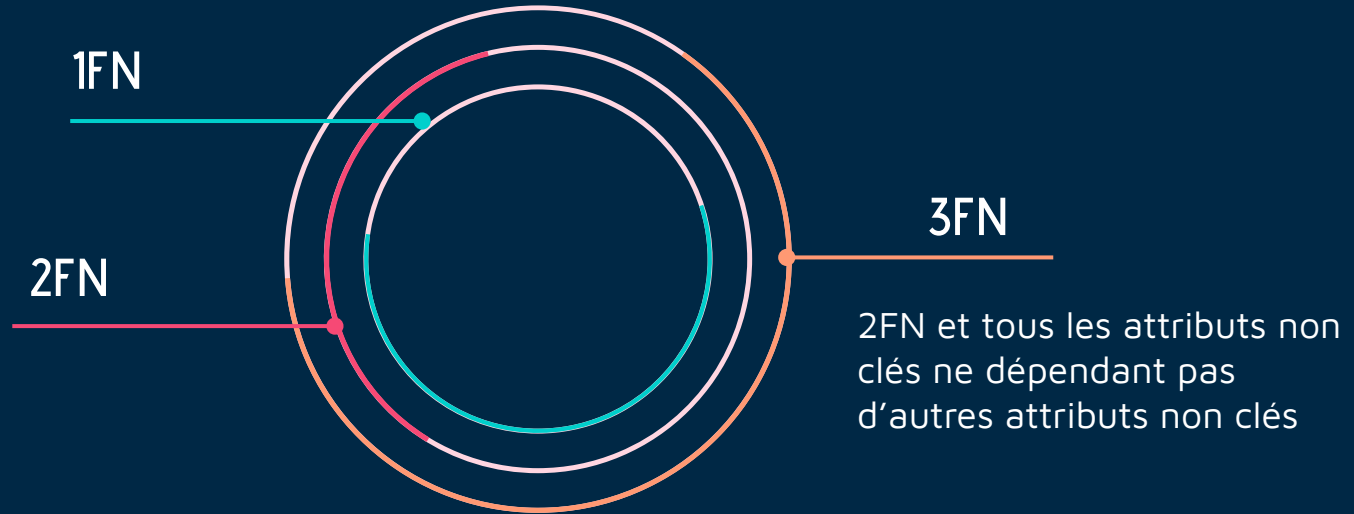
# Formes normales



# 2FN

Restaurant	Evaluation	ComposéDe	Menu	OuvertA	CatégorieResto	Client
<i>mail_resto</i>	<i>id_evaluation</i>	<i>id_commande</i>	<i>mail_resto</i>	<i>mail_resto</i>	<i>mail_resto</i>	<i>id_client</i>
nom_resto	date_evaluation	id_plat	<i>id_plat</i>	<i>jour_semaine</i>	categorie_fille	mdp
tel_resto	avis	Quantité	nom_plat	horraire_midi_debut		nom_client
adresse_resto	note_evaluation		prix_plat	horraire_midi_fin		prenom_client
nbr_place				horraire_soir_debut		
note_resto				horraire_soir_fin		
CommandeClient	Commande	Plat	Allergenes	TypeCommandePossible	RelationCategorie	
<i>id_client</i>	<i>id_commande</i>	<i>id_plat</i>	<i>id_plat</i>	<i>mail_resto</i>	<i>Categorie_fille</i>	
<i>id_commande</i>	date_commande	nom_plat	allergenes	type_commande	Categorie_mere	
nbr_personnes	date_fin	description_plat				
	prix_commande					
	adresse_livraison					
	mail_resto					

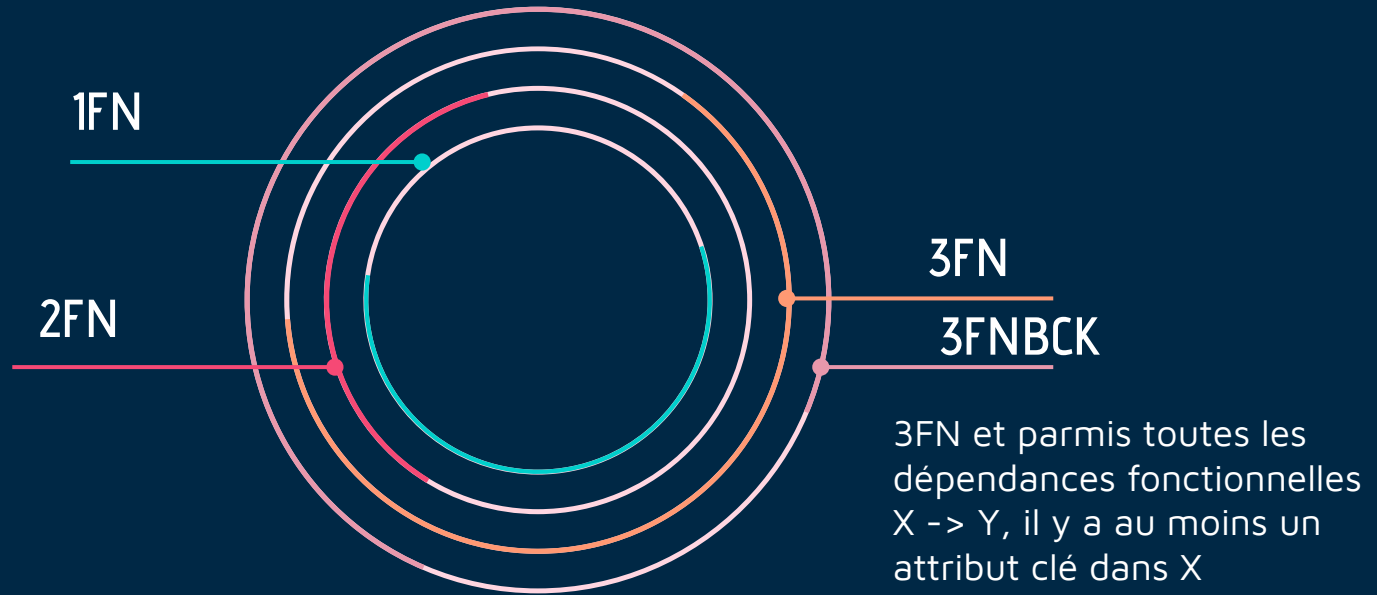
# Formes normales



# 3FN

Restaurant	Evaluation	ComposéDe	Menu	OuvertA	CatégorieResto	Client
mail_resto	id_evaluation	id_commande	mail_resto	mail_resto	mail_resto	id_client
nom_resto	date_evaluation	id_plat	id_plat	jour_semaine	categorie_fille	mdp
tel_resto	avis	Quantité	nom_plat	horraire_midi_debut		nom_client
adresse_resto	note_evaluation		prix_plat	horraire_midi_fin		prenom_client
nbr_place				horraire_soir_debut		
note_resto				horraire_soir_fin		
CommandeClient	Commande	Plat	Allergenes	TypeCommandePossible	RelationCategorie	
id_client	id_commande	id_plat	id_plat	mail_resto	Categorie_fille	
id_commande	date_commande	nom_plat	allergenes	type_commande	Categorie_mere	
nbr_personnes	date_fin	description_plat				
	prix_commande					
	adresse_livraison					
	mail_resto					

# Formes normales



# SQL

Utilisation des commandes SQL pour :

- La Création des tables
- Le respect des contraintes
- Les relations entre les tables

```
create table Menu
(
    mail_resto varchar(100),
    id_plat int,
    nom_plat varchar(100),
    prix_plat float check (prix_plat >=0),
    primary key(mail_resto, id_plat),
    foreign key(mail_resto) references Restaurant(mail_resto),
    foreign key(id_plat) references Plat(id_plat)
);
```



# Parcours des restaurants

**Etape 1 :** Se connecter avec son mail et son mot de passe client

**Etape 2 :** Sélection des catégories de restaurant désirées (ou sélection par défauts des trois catégories préférées)

**Etape 3 :** Affinage éventuel des résultats pour un horaire précis

**Etape 4 :** consultation de la fiche complète (location, type de commande et menu) des restaurants souhaités



# Parcours des restaurants

## Etape 1 :

Nous avons décidé de factoriser cette partie pour des raisons évidentes. On demande simplement le mail du client, on vérifie qu'il existe et on récupère le cas échéant son mot de passe, et on le compare avec le mot de passe donné



## Etape 2 :

La partie délicate est d'étendre la liste des catégories données à la liste complète des catégories engendrées. Pour éviter des accès trop nombreux à la base, on fait une seule sélection et le reste du traitement se fait en java. L'algorithme s'apparente à une exploration d'arbre à partir d'une liste d'arêtes. Ensuite on sélectionne tous les restaurants dont l'une des catégories est dans la liste étendue.

# Parcours des restaurants

CatégorieResto
<i>mail_resto</i>
categorie_fille

RelationCategorie
<i>Categorie_fille</i>
Categorie_mere

### Etape 3 :

Cette partie a engendré une mise à jour des choix de conception, puisque l'on doit faire une comparaison avec les horaires d'ouverture et fermeture, les horaires doivent être un type supportant la comparaison. Ensuite, on reprend la sélection précédente, mais avec un critère de sélection supplémentaire.

# Parcours des restaurants

## OuvertA

*mail\_resto*

*jour\_semaine*

horaire\_midi\_debut

horaire\_midi\_fin

horaire\_soir\_debut

horaire\_soir\_fin

#### Etape 4:

Cette partie est relativement simple, on va simplement chercher toutes les informations voulues pour un restaurant identifié. Une amélioration éventuelle serait de demander les informations à afficher.

# Parcours des restaurants

Restaurant
<i>mail_resto</i>
nom_resto
tel_resto
adresse_resto
nbr_place
note_resto

TypeCommandePossible
<i>mail_resto</i>
type_commande

# Commande

**Etape 1 :** Vérifier le type de commande et sa disponibilité chez le restaurant choisi.

**Etape 2 :** Pour une commande sur place, comparer le nombre de places disponibles au nombre de places à réserver par le client.

**Etape 3 :** Associer un id\_commande à la commande.

**Etape 4 :** Insertion dans CommandeClient.

**Etape 5 :** Insertion dans Commande.



# Commande

Etape 1 : Disponibilité de type commande

Sélectionner \* sous conditions mail\_resto et type\_commande égaux à ceux présentés par le client.

TypeCommandePossible
mail_resto
type_commande

# Commande

## Etape 2 : cas d'une commande sur place

1. Sélectionner le nombre total de places du restaurant choisi.
2. Requête imbriquée pour récupérer le nombre de places occupées à partir de la relation commande et de la relation OuvertA.





# Commande

Etape 3 : générer id\_commande

Sélectionner le nombre  
max des id\_commande  
et l'incrémenter de 1.

Commande
<i>id_commande</i>
date_commande
date_fin
prix_commande
adresse_livraison
mail_resto

# Commande

## Etape 4 et 5

1. Implémentation en java des méthodes nécessaires pour le calcul des attributs de la relation commande.
2. Requête d'insertion pour mettre à jour les relations.

Commande
<i>id_commande</i>
date_commande
date_fin
prix_commande
adresse_livraison
mail_resto

CommandeClient
<i>id_client</i>
<i>id_commande</i>
nbr_personnes

# Les changements de statut des commandes

On traite dans cette fonction tous les cas :  
En effet, le client et le restaurant peuvent changer le statut d'une commande.

Cependant, ils ne peuvent changer que les leurs, et les clients ne peuvent pas mettre les mêmes statuts (un client ne peut pas affecter à une commande le statut "annulée par le restaurant" par exemple).

Enfin, il y a aussi une cohérence à respecter, le statuts "en livraison" ne peut pas précéder "en préparation" par exemple.



**Etape 1 :** Insérer l'évaluation du client dans la table d'évaluation.

**Etape 2:** Faire une mise à jour de la note moyenne du restaurant.

# Evaluation



# Le droit à l'oubli

**Etape 1 :** Dupliquer les données de commande du client avec un nouvel identifiant généré.

**Etape 2 :** Supprimer l'identifiant du client, le nom, le prénom et le mot de passe ainsi que les anciens données.

**Etape 3 :** Supprimer les éventuelles adresses de livraison.



# Démonstrateur

```
*****  
  
    Bienvenue dans GrenobleEats  
  
*****  
  
Identifiant :  
gabriel@gmail.com  
  
Mot de passe :  
chienchat12$  
  
Connexion réussie
```

# Accès du client au différents fonctionnalités

```
Veillez choisir une des fonctionnalités suivante :  
1- Commande  
2- Evaluation  
3- Annuler une commande  
4- Supprimer le compte  
5- Quitter
```

# Pour les restaurants

```
--> ID : 1 | Date de commande : 07/12/2022 22:53:50 | Status : disponible  
--> ID : 78 | Date de commande : 07/12/2022 22:57:50 | Status : en livraison  
--> ID : 65 | Date de commande : 08/12/2022 00:05:50 | Status : attente de confirmation  
--> ID : 32 | Date de commande : 07/12/2022 23:53:50 | Status : en préparation
```



# Conclusion

## Help & Support Icons



## Avatar Icons



## Creative Process Icons



## Performing Arts Icons



# Nature Icons



# SEO & Marketing Icons



