

# GRENOBLE EAT

---

Par Maud Bergonzoli,  
Jorge Luri Vañó, Cédric  
Pauly, Logan Willem





01

**Notre  
organisation**

02

**Analyse du  
problème**

03

**Implémentation**

04

**Démonstration**



# 01

**Notre  
organisation**

# Répartition des tâches

---



**Logan**

- Analyse
- Remplissage des tables
- Droit à l'oubli
- Parcours des restaurants

**BD**



**Maud**

- Analyse
- Remplissage des tables
- Passage de commande

**BD**



**Jorge**

- Interaction utilisateur
- Fonctions de commandes
- Fonctions de parcours
- Connexion avec BDD et SQLite

**API**

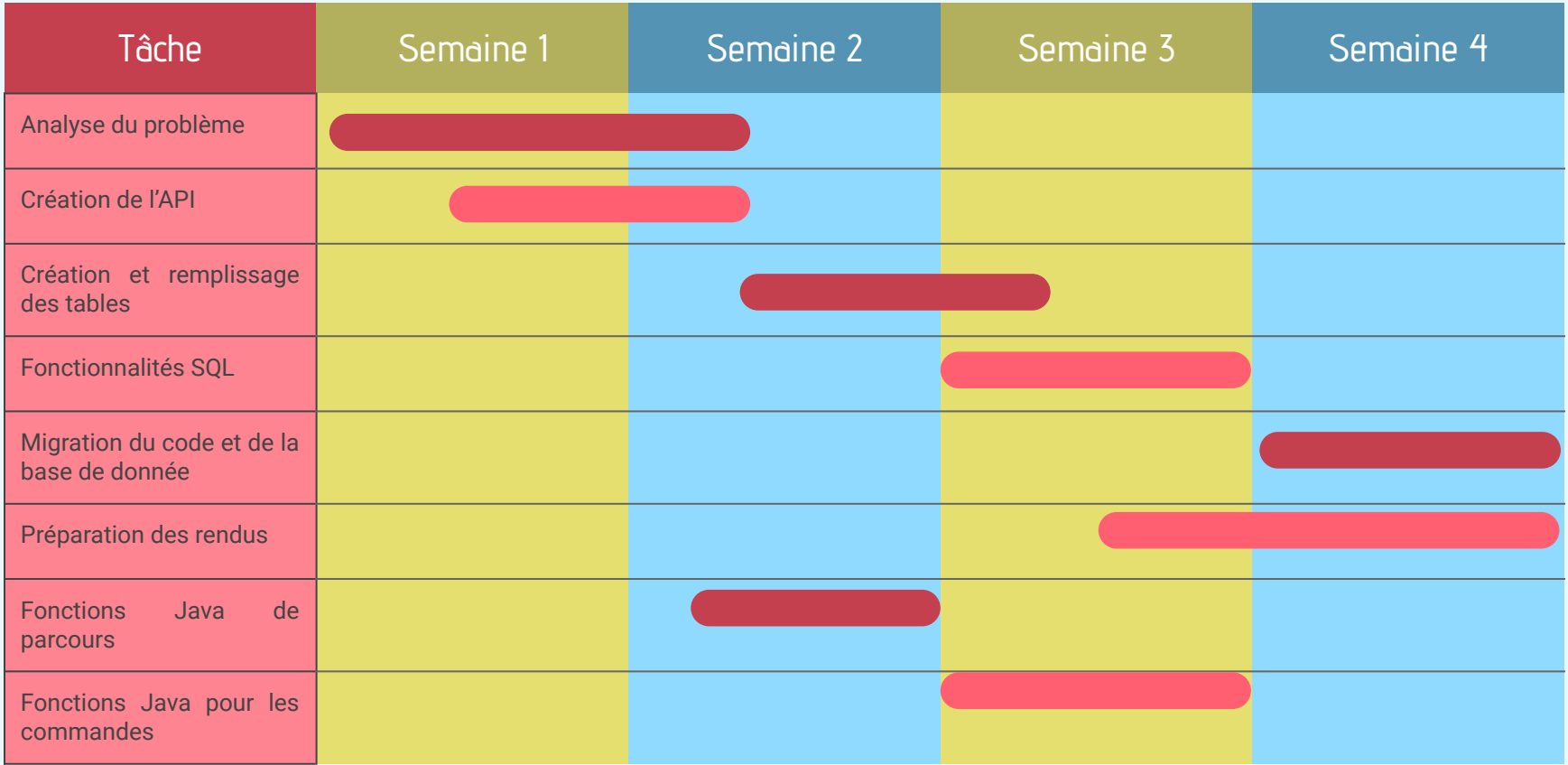


**Cédric**

- Création arbre utilisateur
- Fonction d'évaluation
- Connexion utilisateur

**API**

# Diagramme de Gantt



# 02

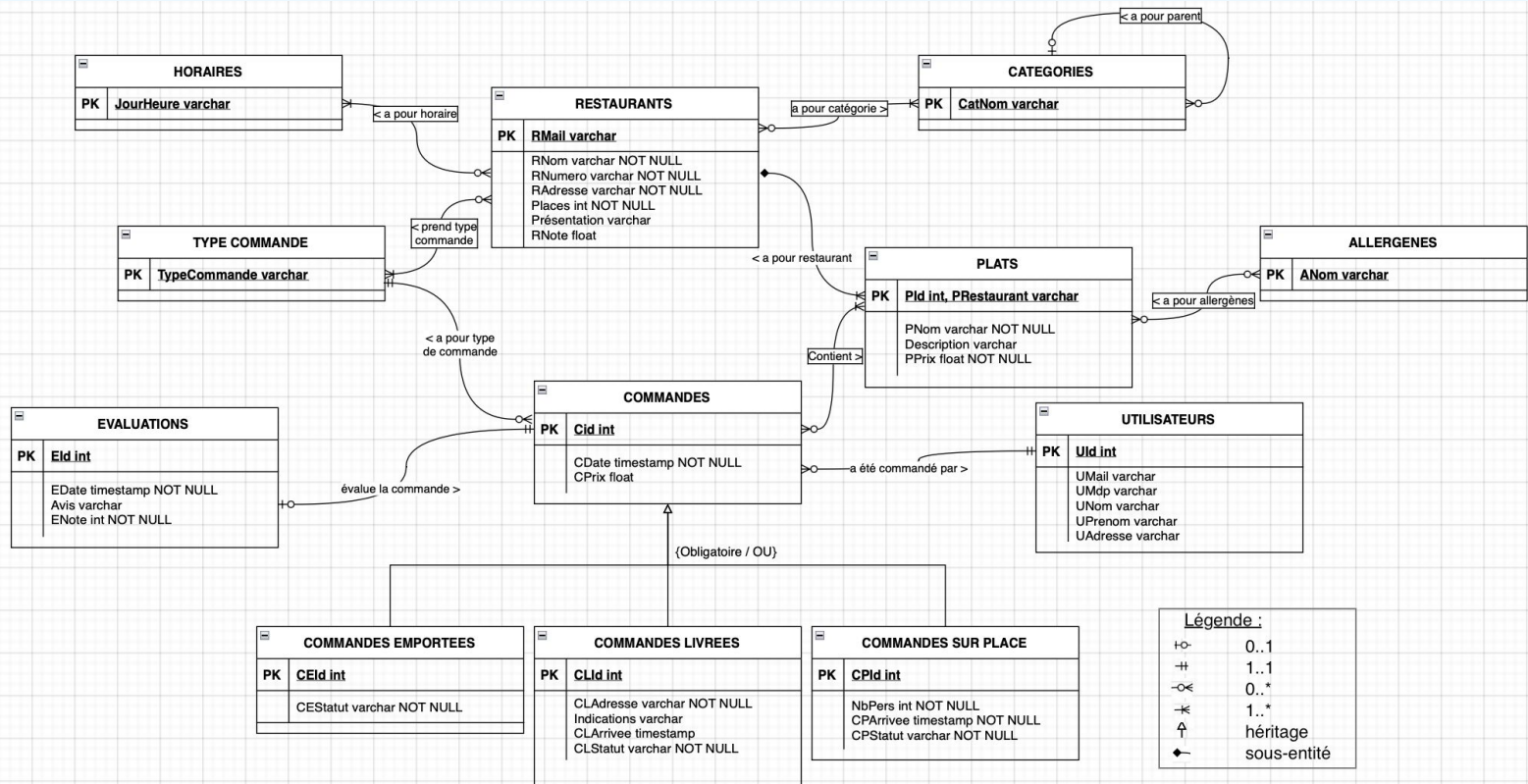
## Analyse du problème



# Contraintes

DF	C. Valeurs	C. Contextuelles	C. Multiplicité
RMail $\rightarrow$ RNom, RNumero, RAdresse, Places, Presentation, RNote	RType $\in$ {livraison, emporter, place}  Places $> 0$  RNote $\in [0, 5]$	$\sum \text{nbPers} \leq \text{Places}$ pour un restaurant et ses commandes associées  $\text{Ext}(\text{CEId}) \cap \text{Ext}(\text{CPId}) \cap \text{Ext}(\text{CLId}) = \emptyset$  $\text{Ext}(\text{CEId}) \cup \text{Ext}(\text{CPId}) \cup \text{Ext}(\text{CLId}) = \text{Ext}(\text{CId})$  CPArrivee $\in$ JourPlage pour un restaurant et ses commandes associées  CDate donne JourPlage pour toute commande  CDate $<$ EDate	RMail $\rightarrow$ JourPlage  RMail $\rightarrow$ TypeCommande  RMail $\rightarrow$ PId  RMail $\rightarrow$ CatNom  (PId, RMail) $\rightarrow$ ANom CId $\rightarrow$ (PId, RMail)
(PId, Restaurant) $\rightarrow$ PNom, PDescription, PPrix	PPrix $> 0$	CDate donne JourPlage pour toute commande  CDate $<$ EDate	RMail $\rightarrow$ CatNom
U_Id $\rightarrow$ UMail, UMdp, UNom, UPrenom, UAdresse			(PId, RMail) $\rightarrow$ ANom CId $\rightarrow$ (PId, RMail)
CId $\rightarrow$ CDate, CPrix	CPrix $> 0$		
CLId $\rightarrow$ CLAdresse, Indications, CLArrivee, CLStatut	CLStatut $\in$ {attente, validée, en livraison, annuleeC, annuleeR, terminée}	EId $\Rightarrow$ CId.statut = {terminée}	CId $\rightarrow$ TypeCommande
CEId $\rightarrow$ CESTatut	CEStatut $\in$ {attente, validée, disponible, annuleeC, annuleeR, terminée}	CDate $\geq$ DateActuelle	CId $\rightarrow$ U_Id
CPId $\rightarrow$ NbPers, CPArrivee, CPStatut	CPStatut $\in$ {attente, validée, annuleeC, annuleeR, terminée} NbPers $> 0$	CPArrivee $\geq$ DateActuelle	CId $\rightarrow$ EId CatNom $\rightarrow$ CatNom
EId $\rightarrow$ EDate, Avis, ENote	ENote $\in [0..5]$	EDate $\geq$ DateActuelle	
	JourPlage $\in$ {LM, LS, MaM, MaS, MeM, MeS, JM, JS, VM, VS, SM, SS, DM, DS }	CType $\in$ TypeCommande pour un CId donné et le RMail associé	

# Diagramme E/A





# Modèle relationnel

---

Entités simples:

- Restaurant(RMail, RNom, RNum, RAdresse, Places, Présentation, RNote)
- Commandes(Cid, CDate, CPrix, **Uid**, **TypeCommande**)
- Utilisateurs(U\_id, UMail, UMdp, UNom, UPrenom, UAdresse)
- Evaluation(Eid, EDate, Avis, ENote, **Cid**)
- Horaires(JourPlage)
- TypesCommande(TypeCommande)
- Categories(CatNom)
- Allergenes(ANom)

Sous-types d'entités (utilisation des références):

- CommandesEmportees(CEid, CESTatut)
- CommandesLivrees(CLid, CLAdresse, Indications, CLArrivee, CLStatut)
- CommandesSurPlace(CPid, NbPers, CPArrivee, CPStatut)

# Modèle relationnel

---

Entité faible:

- Plats(Pid, PRestaurant, PNom, Description, PPrix)

Types d'associations en fonction des cardinalités:

- CategorieParent(CatNom, CatNomMere)
- HorairesRestaurant(RMail, JourPlage)
- CategoriesRestaurant(RMail, CatNom)
- TypesRestaurant(RMail, TypeCommande)
- PlatsCommande(Cid, Pid, PRestaurant)
- AllergenesPlat(Pid, PRestaurant, ANom)



03

Implémentation

# Partie SQL - Remplissage

---

La base de données en quelques chiffres:

- 100 utilisateurs
- 19 restaurants
- 47 plats différents

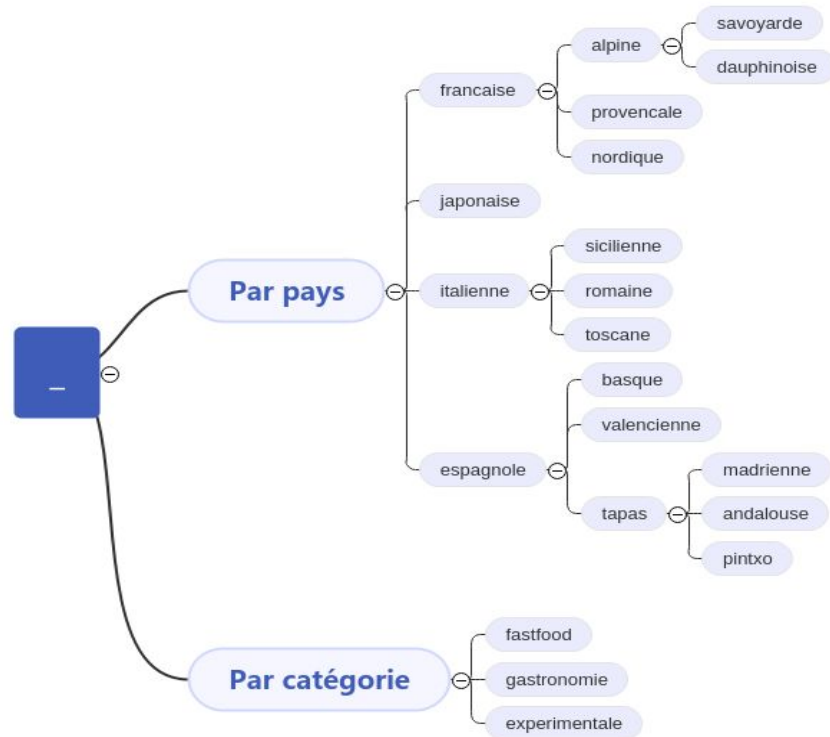
Remplissage des tables à données constantes:

- HORAIRES
- ALLERGENES
- TYPESCOMMANDE

Remarque: Les tables CATEGORIEPARENT, TYPESRESTAURANT, CATEGORIESRESTAURANT, HORAIRESRESTAURANT et ALLERGENESPLAT ont été remplies en conséquence.

# Partie SQL - Remplissage

---



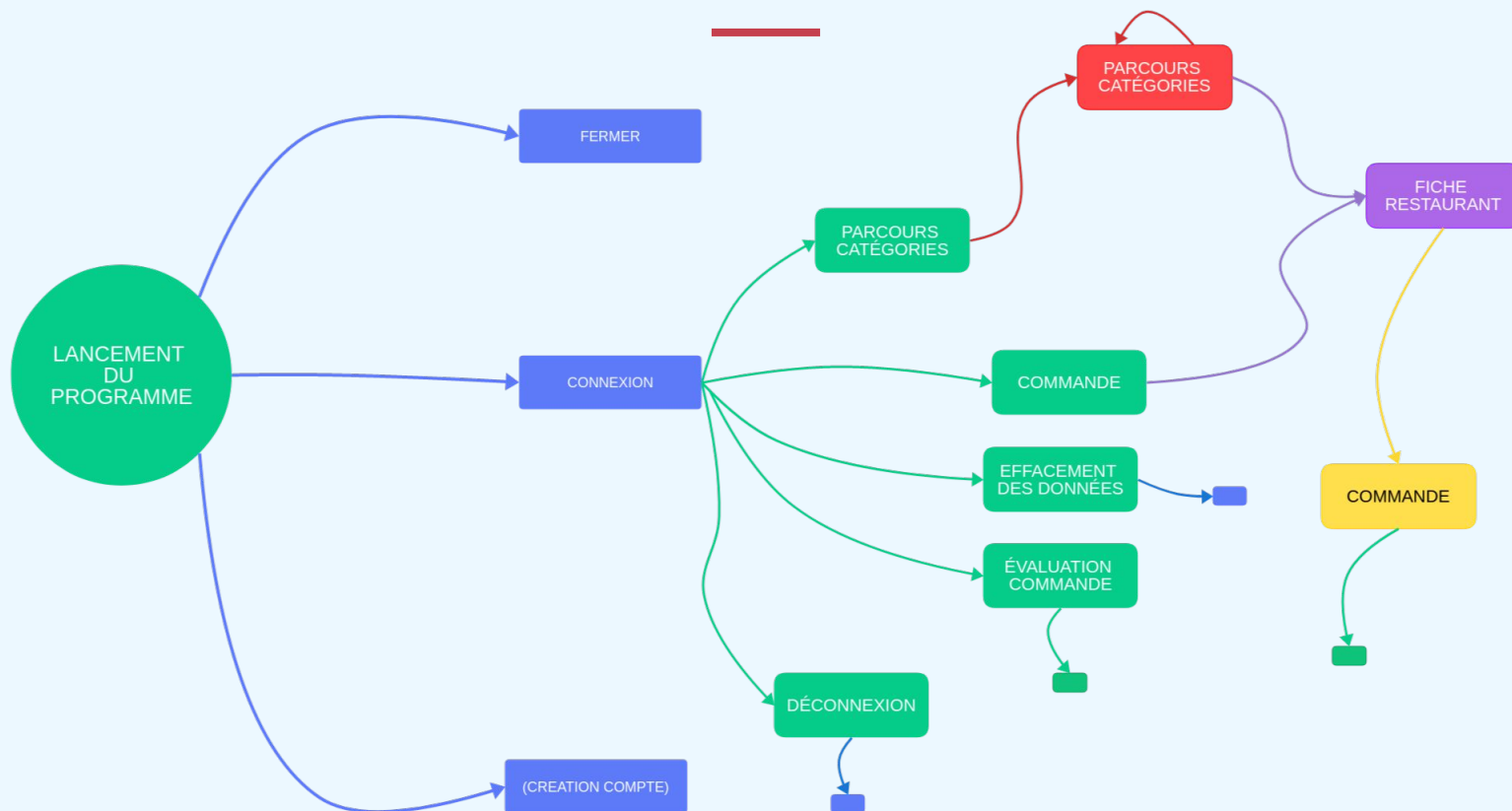
# Partie SQL - Fonctions

---

Scripts SQL implémentant les fonctionnalités:

- Droit à l'oubli: UPDATE de la table UTILISATEURS en changeant les attributs à NULL
- Parcours des catégories: SELECT à la suite pour afficher les catégories et sous-catégories à la façon d'un menu déroulant puis SELECT pour afficher les restaurants suivant les critères d'horaire et de catégorie
- Passage des commandes : INSERT dans COMMANDES, PLATSCOMMANDE et COMMANDESLIVREES puis UPDATE de COMMANDES avec le CPrix

# Partie Java



# Partie Java

---

```
-- -- --  
Bienvenu à GrenobleEAT !
```

```
As-tu un compte ?  
-- -- --
```

- 1) J'ai un compte utilisateur
- 2) Créer un compte
- 3) Quitter l'application

```
Tapez le numéro de la réponse que vous souhaitez : █
```

```
Bienvenue Jorge!
```

```
Que souhaitez-tu faire ?  
-- -- --
```

- 1) Parcourir les restaurants
- 2) Passer une commande
- 3) Supprimer mes données personnelles (Droit à l'oubli)
- 4) Laisser évaluation
- 5) Changer d'utilisateur
- 6) Quitter l'application

```
Tapez le numéro de la réponse que vous souhaitez :
```

```
-- -- --  
Tu vas te régaler !
```

```
Voici la liste des restaurants dans l'ordre décroissant de notes :
```

- 1) Au cochon qui dore
- 2) Bières et tapas
- 3) Croquettes de chez vous
- 4) Gyoza de Ginza
- 5) Il Gorgonzola
- 6) L instant espagnol
- 7) La buena tortilla
- 8) La cuisine du chef
- 9) La fourchette forestière
- 10) La raie gustative
- 11) Voir plus de restaurants

```
0) Retour à l'accueil
```

```
Écris le numéro de ta réponse souhaitée :
```



# Nouvelle implémentation

---



# 04

## Démonstration



# Merci

---

**Avez-vous des  
questions ?**

