Analyse

Dépendances fonctionnelles

 $\underline{emailRest} \rightarrow nomRest$, telRest, adresseRest, horaires, presentation, capaciteMax, noteRest

emailRest, nomPlat, description, prixPlat

 $\underline{\mathsf{idCompte}} \to \mathsf{emailClient}$

 $\underline{emailClient} \rightarrow mdp, \, nomClient, \, prenomClient, \, adresseClient$

 $\underline{\text{dateCommande, heureCommand,idCompte, emailRest}} \rightarrow \text{prixCommande, statut}$

<u>dateCommande</u>, <u>heureCommand</u>, <u>idCompte</u>, <u>emailRest</u>, <u>idPlat</u> → quantitéCommandée

 $\underline{dateCommande,\ heureCommand,idCompte,\ emailRest} \rightarrow adresseLivraison,\ infos,\ heureLivraison$

 $\underline{dateCommande}$, $\underline{heureCommand,idCompte}$, $\underline{emailRest}$ \rightarrow $\underline{nbPersonne}$, $\underline{heureArrivee}$

 $\underline{\text{dateEval, heureEval,emailRest, idCompte}} \rightarrow \text{ avis, note}$

<u>nomCategorieMere</u>

 $\underline{\mathsf{nomCategorieFille}}$

jour, heureOuverture, heureFermeture

Contraintes de valeur

statut \in {attente, validee, disponible, livraison, annuleeClient, annuleeRest} {idLivraison} inter {idSurPlace} = vide nbPersonne <= capaciteMax idCategorie = idCategorieMere U idCategorieFille

Contraintes de multiplicité

emailRest 0..* nomCategorieFille
emailRest 0..* nomCategorieMère
nomCategorieFille 0..1 nomCategorieMère
nomCategorieMère 0..* nomCategorieFille
emailRest 0..* Commande
emailRest 1..* nomPlat
emailRest, idPlat 0..* nomAllergene
nomAllergene 1..* emailRest, idPlat
idCommande 0..1 dateEval, heureEval
contenuCommande 1..* idPlat
idCompte 0..1 emailClient
emailClient 1..1 idCompte

Contraintes textuelles

- Il faut pouvoir classer les restaurants par catégories et/ou par leur note moyenne
- Les recommandations de catégories se basent sur l'historique de commandes des utilisateurs et les notes qui leur sont reliées. En priorité, les recommandations vont concerner les catégories pour lesquelles l'utilisateur a donné la meilleure note moyenne lors de ses commandes. Ensuite, pour départager les catégories se retrouvant à égalité, les recommandations seront triées par ordre décroissant des catégories pour lesquelles il y a eu le plus de notes déposées et ce quel que soit l'utilisateur.
- noteRest = moyenne des notes des commandes
- historique = liste des commandes filtrées par idCompte
- si un client demande à ce que l'on supprime ses données personnelles il est important de conserver les commandes qui ont été effectuées
- si une catégorie n'a pas de mère, c'est une catégorie mère.

Modèle Entité-Association

Types d'entité simple :

 $Restaurant (\underline{emailRest}, nomRest, telRest, adresseRest, horaires, presentation, capaciteMax, noteRest)$

Compte(idCompte)

 ${\sf Client}\;({\sf E}\underline{\sf mailClient},\,{\sf mdp},\,{\sf nomClient},\,{\sf prenomClient},\,{\sf adresseClient})$

Évaluation(dateEval, heureEval, avis, note)

Catégorie(nomCategorie)

Allergene(nomAllergene)

Horaire(jour, heureOuverture, heureFermeture)

Sous-types d'entité

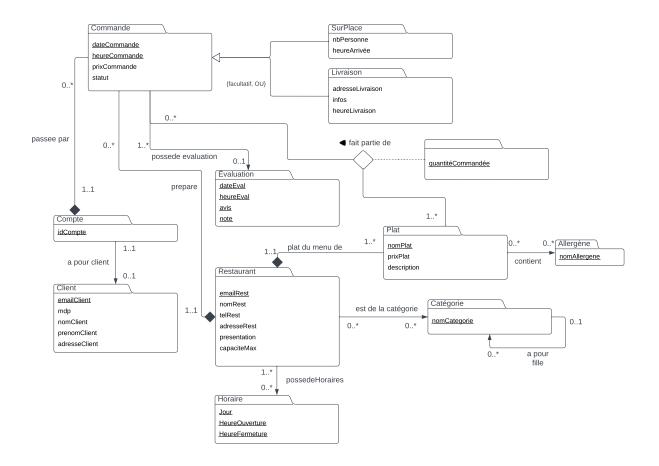
Livraison(adresseLivraison, infos, heureLivraison)

SurPlace(nbPersonne, heureArrivee)

Type d'entité faible

 $Commande (\underline{dateCommande}, \underline{heureCommande}, \underline{prixCommande}, \underline{statut})$

Plat(nomPlat, prixPlat, description)



Passage au relationnel

Types d'entités simples

Restaurant(<u>emailRest</u>, nomRest, telRest, adresseRest, horaires, presentation, capaciteMax, noteRest, nomCategorie) nomCategorie référence Catégorie

 $Commande (\underline{dateCommande,\ heureCommande,\ idCompte\ ,\ emailRest},\ prixCommande,\ statut)$

passeePar = idCompte est non nul et référence Compte

prepare = emailRest est non nul et référence Restaurant

Compte(idCompte)

 $Client(\underline{emailClient},\ mdp,\ nomClient,\ prenomClient,\ adresseClient,\ \underline{idCompte})$

aPourClient = idCompte est non nul et référence Compte

Evaluation(dateEval, heureEval, avis, note)

Catégorie (<u>nomCatégorie</u>)

Horaire(jour, heureOuverture, heureOuverture)

Sous-types d'entité (plusieurs possibilités d'implémentation) : on choisit le référencement

 $SurPlace (\underline{dateCommande,\ heureCommande,\ idCompte\ ,\ emailRest},\ nbPersonne,\ heureArrivee)$

<u>dateCommande, heureCommande, idCompte, emailRest</u> de SurPlace référence <u>dateCommande, heureCommande, idCompte,</u> emailRest de Commande

 $Livraison (\underline{\textbf{dateCommande, heureCommande, idCompte, emailRest}, adresse Livraison, infos, heureLivraison)$

<u>dateCommande, heureCommande, idCompte, emailRest</u> de Livraison référence <u>dateCommande, heureCommande, idCompte, emailRest</u> de Commande

Entité faible

Plat(<u>emailRest</u>, <u>nomPlat</u>, prix, description)

platDuMenuDe = emailRest de Plat est non nul et référence emailRest de Restaurant.

Vérifier que tout restaurant ait au moins un plat.

Associations

ApourFille(idCatégorieMère, idCatégorieFille)

idCatégorieMère et idCatégorieFille référencent Catégorie(idCatégorie)

faitPartieDe(dateCommande, heureCommande, idCompte, emailRest, idPlat, qu antitéCommandée)

dateCommande, heureCommande, idCompte , emailRest référence Commande idPlat référence Plat

Vérifier qu'une commande ait au moins un plat.

contient(emailRest, nomPlat, nomAllergene)

 $possede Horaires (\underline{emailRest, jour, heure Ouverture, heure Fermeture})$

possedeEvaluation(dateCommande, heureCommande, idCompte, emailRest, dateEval, heureEval)

