

---

## **Projet BDR : The Beer Garden**

Erica Akoumba, Olivier D'Ancona, Jean-François Pasche

26 novembre 2021

## **The Beer Garden : Modélisation conceptuelle**

### **Table des matières**

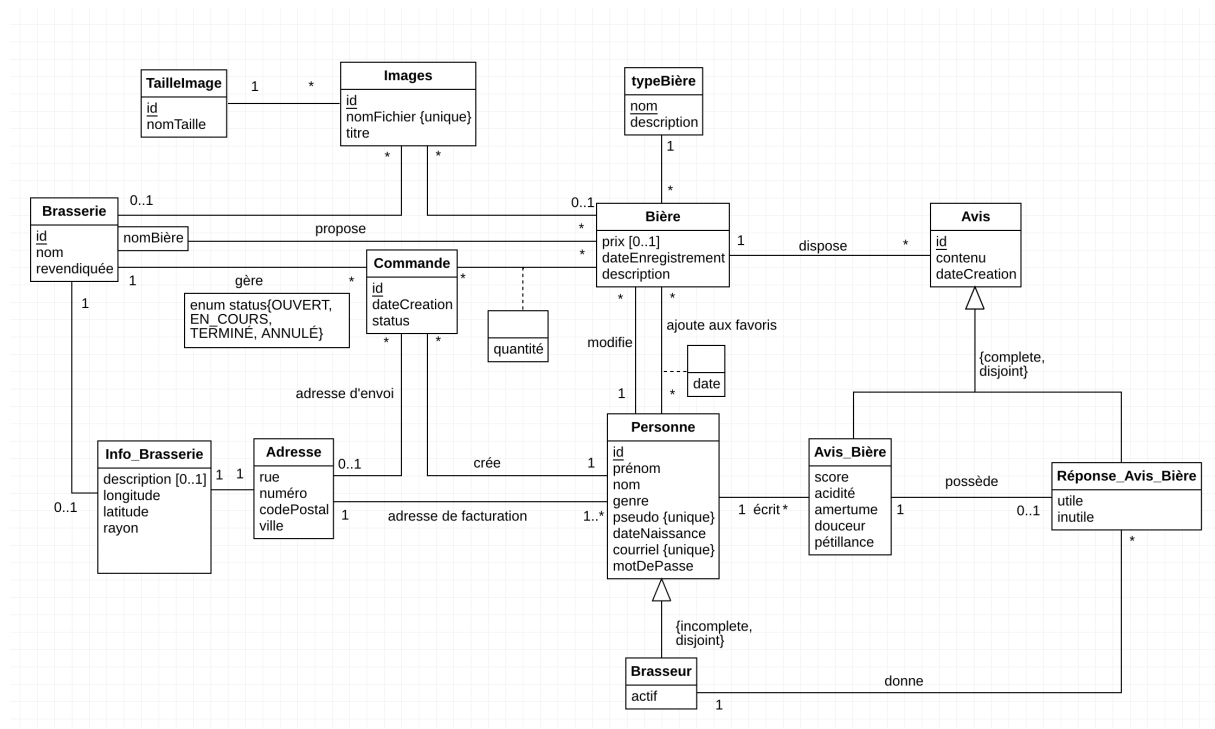
1. Introduction
2. Modélisation
  - 2.1. Schéma EA
  - 2.2. Contraintes d'intégrité
3. Choix de conception
  - 3.1. Explications complémentaires
  - 3.2. Remarques sur le cahier des charges
4. Conclusion

### **Introduction**

Voici la suite de la modélisation de notre application beergarden. Dans ce document, nous détaillons le fonctionnement de notre application. Premièrement, nous constituons le schéma entité-association qui représentera l'ensemble des types d'entités ainsi que leurs relations. Ensuite, nous expliquerons en quelle mesure notre schéma diffèrera du cahier des charges. Finalement, nous listerons les contraintes d'intégrité inhérentes à notre modélisation.

## Modélisation

### Schéma EA



**Figure 1:** Modélisation de “The Beer Garden”

## Contraintes d'intégrité

## Personne

- La date de naissance doit correspondre à un âge de plus de 18 ans.

## Brasseur

- Un brasseur désactivé ne peut pas répondre à un avis de bière.

## Bière

- Le prix doit être supérieur ou égal à 0 lorsqu'il est défini

### **Avis**

- La date de création doit être inférieure à la date actuelle.
- La date de création doit être supérieure à la date de création d'une bière.

### **Avis\_Bière**

- Le score doit être compris entre 2 et 10.
- L'acidité, l'amertume, la douceur et la pétillance ont des valeurs comprises entre 1 et 10.

### **Réponse\_Avis\_Bière**

- utile et inutile doivent être supérieurs ou égaux à 0.
- Une réponse à un avis ne peut pas avoir une date inférieure à l'avis concerné.
- Seuls les brasseurs actifs peuvent donner une réponse.

### **Bière - Personne (favoris)**

- La date d'ajout dans les favoris doit être inférieure à la date actuelle.
- La date d'ajout dans les favoris doit être supérieure à la date de création de la bière.

### **Bière - Commande**

- La quantité doit être supérieure à 0.
- La date de création de bière doit être égale ou inférieure à la date actuelle.

### **Brasserie**

- Un brasseur ne peut pas revendiquer une brasserie déjà revendiquée.

### **Adresse**

- Le numéro de rue doit être supérieur à 0.
- Le numéro postal doit être supérieur à 0.
- Si un utilisateur ajoute une adresse d'envoi, elle doit être différente de l'adresse de facturation.

### **Commande**

- La date de commande doit être inférieur ou égale la date actuelle.
- La date de commande doit être supérieure à la date de création de la bière.

**Info\_Brasserie**

- Les valeurs de longitude et de latitude sont comprises dans l'intervalle  $[-180, 180]$ .
- le rayon doit être supérieur ou égal à 0.

**Choix de conception****Explications complémentaires****Brasserie**

- `revendiquée` est un champ booléen. `false` signifie que la brasserie n'a pas été revendiquée.
- Lorsqu'un brasseur revendique une brasserie, les droits de modification des bières de cette brasserie lui sont transférés par une modification de la relation `modifie` entre `Personne` et `Bière`. Le champ `revendiquée` de l'entité `Brasserie` passe à `true`.

**Remarques sur le cahier des charges**

Quelques adaptations ont dû être faites par rapport au cahier des charges. La possibilité de supprimer un compte a été remplacée par la possibilité de désactiver un compte. Cela évitera la suppression des commentaires d'un client qui supprimerait son compte. En revanche, un brasseur peut supprimer les informations contenues dans `Info_Brasserie` sans que cela n'affecte les commandes passées dans une brasserie, car celles-ci sont liées à l'entité `Brasserie`, dont les instances ne pourront pas être supprimées.

Enfin, le rôle d'administrateur a été retiré, car il n'apporte pas grand chose du point de vue de la base de donnée mais demanderait beaucoup de travail du point de vue de l'interface utilisateur.

**Conclusion**

La création du modèle EA nous a mené à remettre en cause certains aspects du cahier des charges, plus précisément la manière de supprimer un compte. A ce stade, la question se pose de séparer l'entité `Personne` en deux entités `Personne` et `Info_Personne`, avec dans la première le pseudonyme et le mot de passe et dans la seconde toutes les autres informations. Cela permettrait de lier `Personne` à `Commande`, `Avis_Bière`, `Bière` et non `Info_Personne`. Ainsi, on pourrait supprimer les informations personnelles d'un utilisateur sans affecter ses relations avec les différentes entités. A voir si la solution proposée est pertinente.