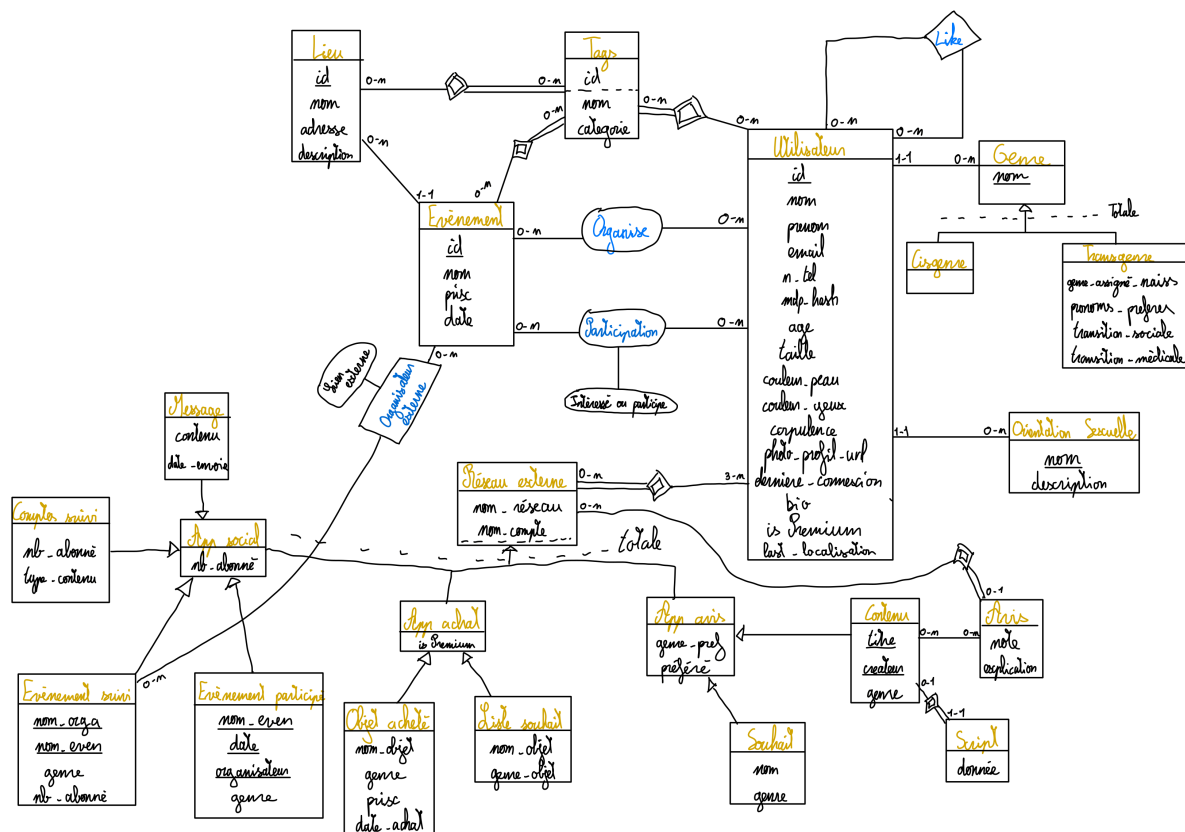


Rapport de Bases de Données

1) Modélisation de la base de donnée :

1.1) Schéma E/R :



Contrainte externes :

- L'âge d'un utilisateur doit être supérieure ou égale à 18.
- Genre assigné naissance peut être **NULL** si l'utilisateur ne veut pas le renseigner.
- Pour le tag, l'utilisateur, l'évènement ou le lieu peuvent mettre n'importe quels noms si la catégorie est autre. Sinon, le nom doit déjà faire partie de la table.
- Pour le genre, l'utilisateur peut décider si la précision de **transgenre** ou de **cisgenre** est affiché dans le profil ou non.
- Si l'utilisateur a choisi un genre *Homme* ou *Femme*, *Cisgenre* ou *Transgenre* et que son orientation sexuelle est *Hétérosexualité* ou *Homosexualité*, alors il n'aura pas à préciser par quel genre l'utilisateur est attiré.
- Si, dans la table Évènement, le prix est **NULL**, alors cela signifie que l'évènement est gratuit.
- Si aucun organisateur n'est renseigné pour un évènement, alors il faut impérativement qu'il existe un lien externe à l'application qui redirige vers toutes les informations de l'évènement car il sera automatiquement supposé externe.
- Pour le nombre d'évent créer dans la table d'utilisateur, il doit être inférieure ou égale à 3 si un utilisateur n'a pas un compte premium.
- Lorsqu'un évènement a un organisateur externe, alors il faut que la date de l'évènement de l'organisateur externe soit égale à la date de l'évènement de l'appli.

1.2) Explication de la modélisation :

Nous allons séparer en 3 parties notre modélisation :

- **La partie Utilisateurs**, contenant dans notre modélisation E/R :
 - *Utilisateur*
 - *Genre* et ses sous-relations
 - *Orientation sexuelles*
 - *Tags* (qui est aussi présent dans la partie **Évènements**)
- **La partie Évènements**, contenant dans notre modélisation E/R :
 - *Évènement*
 - *Lieu*
 - *Tags* (qui est aussi présent dans la partie **Utilisateurs**)
- **La partie Réseau Externe**, contenant dans notre modélisation E/R :
 - *Réseau externes*, qui sera représenté par 3 sous relations :
 - *Application Social*
 - *Application Achat*
 - *Application Avis*

1.2.1) La partie Utilisateurs :

Dans la partie **Utilisateurs**, nous retrouvons toutes les informations utiles sur l'utilisateur de l'application tel que son nom, son prénom, son âge, le fait qu'il possède un abonnement premium ou non, etc... Un utilisateur sera différencié par un *id*.

On retrouve aussi le genre que l'utilisateur fournira qui possède une relation totale avec *cisgenre* et *transgenre* :

- Si l'utilisateur est *cisgenre*, alors aucune information supplémentaire est demandé.
- Si l'utilisateur est *transgenre*, alors l'utilisateur pourra fournir :
 - Le genre assigné à la naissance (s'il le souhaite bien sûr),
 - Le pronom qu'il préfère qu'on utilise pour désigner l'utilisateur.
 - Si la transition est sociale ou non, c'est à dire si le fait que l'utilisateur est transgenre est une information publique ou non.
 - Si une transition médicale a été effectué.

L'utilisateur fournira aussi son orientation sexuelle, qui sera défini par un nom et une description pour que lorsqu'un utilisateur tombe sur un profil avec une orientation sexuelle, il puisse cliquer sur l'orientation et aura accès à sa description.

Dans certains cas, lorsqu'il aura choisi son orientation sexuelle, il devra préciser par quel(s) genre(s) l'utilisateur est attiré.

Un utilisateur peut aussi fournir des tags (*mots-clefs*) qui sera associé à son profil et qui permettra la recommandation à des évènements ou à d'autres utilisateurs.

On retrouvera aussi une *table de jointure* qu'on nommera *Like* et qui représentera un like entre deux utilisateurs. On peut le voir ainsi :

- Si un tuple **(A,B)**, avec **A** et **B** qui représente l'identifiant d'un utilisateur **disjoint**, alors cela signifie que l'utilisateur **A** a liké l'utilisateur **B**.
- Il y'a un **match** lorsque 2 utilisateurs se sont likés mutuellement.

1.2.2) La partie Évènements :

Dans la partie **Évènements**, on retrouve les informations de base d'un événement tels que son nom, le prix et la date. Chaque événement peut avoir ses propres tags qui permettront de trouver rapidement des personnes pouvant potentiellement être intéressé par l'évènements.

Un tuple de la table *Évènement* peut posséder un organisateur étant un utilisateur de l'application, ou sinon un utilisateur externe et donc provenant d'un *Réseaux Externes*, partie traitée plus tard.

Chaque événement est rattaché à 1 lieu qui lui même pourra avoir des tags qui permettront de trouver encore plus rapidement des utilisateurs possédant le même tags que le lieu et donc pouvant être intéressé par l'évènement.

Un tuple de la table *Évènement* aura une relation avec la table *Utilisateur* via une *table de jointure* Participation qui aura comme attribut l'*id* de l'évènement, l'*id* de l'utilisateur et l'état qui pourra prendre 3 valeurs :

- L'état sera **NULL** si une invitation est envoyé et que l'utilisateur n'y a pas répondu.
- L'état sera **Intéressé** si l'utilisateur a répondu qu'il était intéressé par l'évènement mais que sa participation n'est pas garantie.
- L'état sera **Participe** si l'utilisateur a répondu positivement à l'invitation.

Si aucun tuple avec un utilisateur donné est présent dans la table, cela signifie que soit l'utilisateur n'a pas reçu d'invitation pour cet événement ou que l'utilisateur a refusé l'invitation à l'évènement.

Lorsqu'un utilisateur cliquera sur un lieu, via notre application, une description du lieu s'affichera avec ses tags associés.

Le fonctionnement des tags est similaire à la partie **Utilisateurs**.

1.2.3) La partie Réseaux Externes :

Dans la partie **Réseaux Externes**, les informations concernant toutes les *sources externes* que nous allons utiliser pour créer un algorithme de *match potentiel*, comme le souhaite le projet en s'immerçant dans l'empreinte digitale de l'utilisateur. Pour une raison de clarté, nous avons décidé de séparer les "*réseaux externes*" en trois catégories :

- **Une application sociale :**
 - Elle représentera un réseau social, tel que **Facebook, Twitter, Instagram**, etcc où l'on pourra récupérer tous les événements suivi ou les événement auquel l'utilisateur a déjà participé.
 - On pourra récupérer aussi les comptes suivis par l'utilisateur dans ce réseau social, et son nombre d'abonné.
 - On pourra utiliser cette table pour trouver un *match potentiel* en comparant le nombre d'abonner de l'utilisateur, les comptes suivis, les événements suivis et les événements dans lesquels les 2 utilisateurs ont participé.
- **Une application d'achat :**
 - Elle représentera une plateforme d'achat, tel **qu'Amazon, AliExpress, H&M**, etcc où l'on pourra récupérer l'historique d'achat de l'utilisateur et la liste de souhaits de l'utilisateur.
 - On pourra utiliser cette table pour trouver un *match potentiel* en regardant les objets similaire acheté par les utilisateurs et à quel date. Par exemple, si les 2 dates d'achats sont proches, alors il ont plus de chance d'être recommandé l'un à l'autre.
 - On pourra aussi savoir s'il possède un compte premium à la plateforme. Par exemple s'il possède Amazon Prime dans le cas d'Amazon.
- **Une application avis :**
 - Elle représentera une application où l'on pourra donner un avis, tel que **LetterBox, Spotify, IMDb**, etc... où l'on pourra récupérer les avis donné à un *article* tel qu'une chanson, un film, une série, etc...
 - On aura aussi accès au contenu et à ses données, un script pour un film, des paroles pour une chanson, etc...
 - On pourra utiliser cette table pour trouver un *match potentiel* en comparant les avis donné à un genre de film, ou à un même film et l'occurrence d'avis pour un même genre de film.

- On pourra aussi récupérer l'article préféré de l'utilisateur dans la plateforme et son genre préféré aussi.

Ces trois catégories vont former une relation totale avec la table **Réseau Externe** dans notre schéma E/R. Si un événement provient d'un Réseau Externe, alors l'organisateur sera représenté par un lien externe au site. On ne pourra pas avoir un organisateur externe et un organisateur interne à l'application simultanément.

1.3) Schéma relationnel :

```
Utilisateur(idU, nom_genre*, type_genre*, nom_orientation*, nom, prenom, email, numero_tel, mdp_hash, age, tai
GenreUtilisateur(idU*, nom_genre*, type_genre*, genre_assigne_naiss, pronom_preferes, transition_sociale, trans
Genre(nom_genre, type_genre)
Orientation(nom_orientation, description_orientation)
Aime(idU1*, idU2*)
```

```
Evenement(idE, idL*, nom, prix, description_event, date_event)
Lieu(idL, nom, adresse, description_lieu)
Organisateur(idU*, idE*)
Participation(idU*, idE*, type_participation)
```

```
Tag(idT, nom, categorie_tag)
TagUtilisateur(idT*, idU*)
TagLieu(idT*, idL*)
TagEvenement(idT*, idE*)
```

```
ReseauExterne(idRE, idU*, nom_reseau, pseudo)
```

```
AppSocial(idAS, idRE*, nb_abonne)
MessageAppSocial(idAS*, contenu, date_envoi)
CompteSuivi(idAS*, pseudo, nb_abonne, type_contenu)
EvenementSuivi(idAS*, nom_orga, nom_event, genre, nb_abonne)
EvenementPasse(idAS*, nom_event, date_event, organisateur, genre)
OrganisateurExterne(nom_orga*, nom_event*, idE*, date_event, lien_externer)
```

```
AppAchat(idAAc, idRE*, isPremium)
ObjetAchete(idAAc*, genre, prix, date_achat)
ListeSouhait(idAAc*, nom_objet, genre_objet)
```

```
AppAvis(idAAv, idRE*, genre_pref, prefere)
Contenu(titre, createur, donnee)
AppAvisContenu(idAAv*, titre*, createur*)
Avis(idAAv*, titre*, createur*, note, explication)
Souhait(idAAv*, nom, genre)
```

1.3.1) Contrainte d'intégrité :

- Les **clefs primaires** sont soulignées et les **clefs étrangères** sont indiquées par : *.
- Toutes les clefs étrangères possèdent le comportement **ON DELETE CASCADE**.
- L'âge d'un utilisateur doit être supérieur à 18.

1.3.2) Choix par rapport au schéma E/R :

Nous avons décidé d'implémenter plusieurs types de requêtes, avec certaines qui pourront servir pour l'algorithme de matching.

- Pour le **Tag**, nous avons décidé de créer une table **Tag** et d'autres tables *de jointure*. Cela ne respecte pas vraiment la contrainte d'entité faible mais sans cela, il serait plus complexe de lier plusieurs tags à nos utilisateurs, événements ou lieux. Nous avons donc préférés utiliser des tables de jointures.
- Pour toutes les tables qui possédait seulement une relation classique (seulement un trait) :
 - Si la relation était un 1-n ou 0-n, alors dans la table de relation "1", nous avons mis la clef primaire de la table de relation "n".
- Pour toutes les tables qui possédait une relation avec un *contexte* entre les deux (par exemple Organise) :
 - Nous avons créer des tables de jointure où nous mettons les clefs primaires des 2 tables en relations.
- Pour toutes les tables qui possédait des relations :
 - Totale, nous avons décider de mettre tous les attributs des tables filles chez la mère et de caractérisé le cas par un attribut *type_nom-table*.
 - Dans le cas de Réseau Externe, nous avons préférés plutôt faire l'inverse en ne mettant que la clef primaire de la table réseau dans les tables filles.
 - Non totale, ce qui représente seulement toutes les filles de la table Réseau Externe, nous avons préférés mettre dans leur table fille les clefs étrangère des tables fille de Réseau Externe.
 - Elles ne possèdent pas de relation totale, car un utilisateur peut avoir un compte dans l'une de ces applications mais n'avoir jamais fait d'activité dessus.

2.1) Les requêtes statistique :

Ces requêtes vont plutôt servir à établir des statistiques sur les données que nous possédons sur nos utilisateurs, événements, etc..

Ceci peut être utile car, ayant accès à toutes l'empreinte numérique de notre utilisateur grâce à son consentement, nous pouvons nous servir, à mauvaise escient, de ces données pour les revendre à d'autres entreprise dans un but lucratif ou pourraient nous servir pour améliorer nos services par exemple en recommandant au utilisateur d'utiliser ses tags car plus populaire, etc...

Les requêtes statistiques sont :

- Les événements organisés par des utilisateurs dont les centres d'intérêt sont liés au lieu ou au type d'événement.
- Les utilisateurs n'ayant pas participé à un évènement ayant le tag 'Hit Me Hard And Soft'.
- La moyenne des prix des événements organisé par un utilisateur externe.
- Les lieux ayant le même nombre d'événement produit et le même nombre de tags.
- Tous les utilisateurs qui ont dépensé plus de 100 euros dans des objets de la catégorie 'Running'.
- Tous les événements les plus chères d'un lieu.
- La taille maximale et la moyenne d'âge des utilisateurs ayant au moins un tag de la catégorie 'Heureux'
- Le prochain jour avec aucun évènement du lieux avec le plus d'événement organisé dans l'application :
 - Pour celle-ci, la récursion est utilisé et permet de s'arrêter dès qu'une date est disponible. Sans cela, nous aurions du calculer à chaque fois en modifiant la date d'un jour.
- Les utilisateurs qui, pour au moins un réseau externe, ont un pseudo qui commence par leur prénom (Ceci peut aussi être catégorisé dans les requêtes de matching)
- Pour chaque utilisateur, son réseau externe où il a le plus d'abonnés ou NULL s'il n'a pas d'app social.
- Les utilisateurs qui ont un compte sur au moins 20% des réseaux externes totales

- Les utilisateurs ayant participé à des événements payant :
 - Pour celle-ci, étant donné que les prix peuvent être NULL et donc peuvent signifier qu'un événement est gratuit, et aussi que 0 signifie que l'événement est gratuit, le résultat n'est pas le même si des NULL sont présent dans les tables.
- Tous les utilisateurs qui ont donné au moins deux avis positifs successifs (dans l'ordre des titres) sur des contenus du même genre, avec une note qui augmente d'un avis à l'autre.

2.2) Les requête de matching :

Ce sont les requêtes qui vont, ou pourraient servir, pour notre algorithme de matching. Ce sont généralement des requêtes qui renvoie des paires d'utilisateur qui les regrouperont très souvent sur des points communs.

Les requêtes de matching sont :

- Les paires d'utilisateur ayant eu un match.
- Tous les paires d'utilisateurs et le mots dominant dans les contenus où ils ont déposé un avis positif (donc plus de 5) :
 - Pour celle-ci, nous utilisons le fenêtrage, car cela nous permet d'ordonner les mots dominant par un rang, on récupère celui au rang 1 et on le compare avec celui de l'autre utilisateur.
- Les utilisateurs qui sont à 3 niveaux de *like* d'un utilisateur donné:
 - Pour celle-ci, nous utilisons la récursion car elle permet de répondre efficacement à la question.
- Les paires d'utilisateurs qui aiment des personnes ayant au moins un même tag utilisateur.
- Pair d'utilisateurs qui suivent au moins un même type de contenu.
- Les paires d'utilisateurs qui partagent au moins un tag commun, qui fréquentent les mêmes lieux et où ces lieux possèdent également ce même tag commun.

3) Algorithme de matching :

3.1) Application dans notre contexte

Notre algorithme va se baser sur plusieurs critères :

- Les tags en commun.
- L'orientation sexuelle.
- La distance géographique entre nos 2 utilisateurs.
- Si les 2 utilisateurs ont le même mots qui apparaissent plusieurs fois dans des chansons où ils ont déposés un avis positifs.
- Si les 2 personnes sont considérés comme *aigri*, alors ils sont plus compatibles.
- Si les 2 personnes sont considérés comme *fan*, alors ils sont plus compatibles.
- Les événements qu'ils ont déjà fait en commun.

4) Contraintes externes et questions éthiques et légaux :

4.1) Contraintes externes non implémenté :

- Le fait d'envoyer une notification lorsqu'un utilisateur se trouve à proximité :
 - Il faudrait que l'application mettent à jour à chaque fois dans la base de données la dernière position de l'utilisateur. Il peut ensuite peut-être faire une requête pour récupérer les utilisateurs proche pour ensuite leur renvoyer une notification.

- Seulement un utilisateur premium peut organiser plus de 3 événements. Aussi, il peut accéder à son historique s'il est premium (historique de like ou d'événement). Aussi, ils peuvent avoir accès au match potentiel qui était présent à ces événements.

4.2) Questions éthiques et légaux :

4.2.1) Collecte et traitement de données sensibles :

- Nous stockons beaucoup d'information personnelles très sensibles telles que :
 - L'identité, l'orientation sexuelle, le genres, la transition médicales/sociales, photos, localisation, préférences intimes.

Cela pose des risques éthiques. La collecte et la gestion de ces données doivent impérativement respecter la vie privée des utilisateurs.

Sinon, elle peut être utilisée pour créer de la discrimination, du harcèlement ou des atteintes à la dignité.

4.2.2) Localisation en temps réel :

Nous avons accès à la position réelle, enfin en théorie, de notre utilisateur. Si une personne malintentionnée réussit à accéder à notre base de données, cela peut poser problème.

4.2.3) Profilage et catégorisation :

Le projet classe les utilisateurs selon genre, orientation, préférences, etc. On a un risque de stigmatisation, de catégorisation rigide ou biaisée ou même de l'exclusion sociale.

Les questions à se poser pour être en conformité sont :

- **La transparence** : Expliquez-vous clairement quelles données vous collectez et pourquoi ?
- **La sécurité** : Quelles mesures techniques et organisationnelles sont en place pour protéger les données ?
- **La limitation** : Est-ce qu'on collecte uniquement les données nécessaires à l'application ?
- **La durée** : Combien de temps sont stockées ces données ? Est-ce qu'on les garde indéfiniment ?
- **Le droit des utilisateurs** : Comment les utilisateurs peuvent-ils accéder, modifier ou supprimer leurs données ?
- **La gestion des incidents** : Avez-vous un plan en cas de fuite de données ?
- **La conformité aux lois locales** : L'application est-elle conforme aux réglementations du pays cible ?

Le projet *Le Big Match*, en utilisant une quantité importante de données personnelles pour organiser des rencontres et créer des événements, soulève des questions éthiques majeures.

En effet, si les algorithmes de mise en relation se basent uniquement sur les comportements individuels, sans aucune considération éthique ou morale, il devient possible de connecter des personnes pour de mauvaises raisons.

Par exemple, cela pourrait aboutir à la création d'événements regroupant des individus partageant des opinions extrêmes, voire haineuses, ce qui risquerait d'encourager des dynamiques toxiques, dangereuses ou discriminatoires.