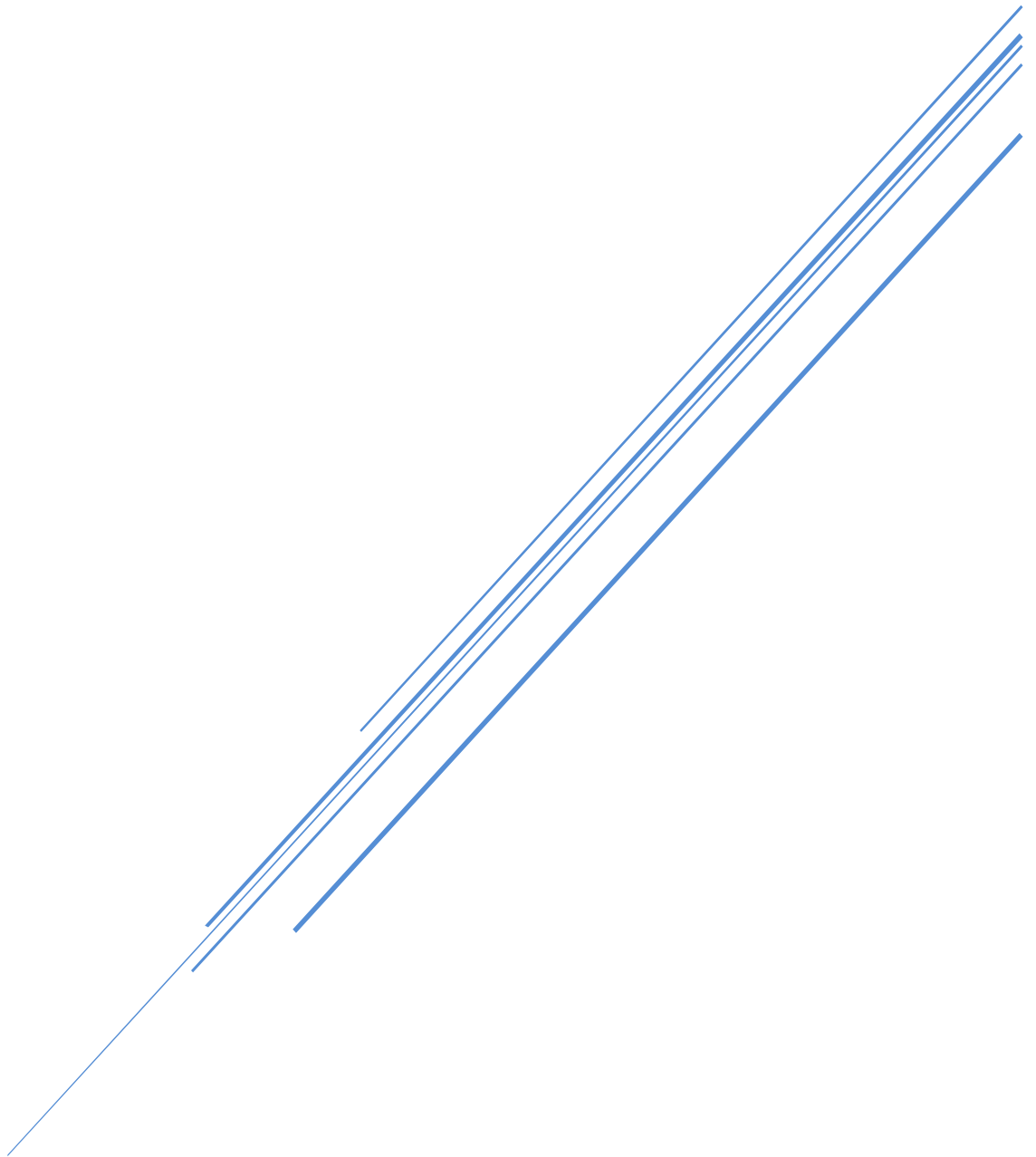


MODÉLISATION CONCEPTUELLE

Groupe 4 – Marengo – Friedli - Marzullo



BDR

Table des matières

1	Introduction	2
2	Modélisation conceptuelle	2
2.1	Choix de modélisation	2
2.2	Contraintes d'intégrité	3
2.3	Schéma.....	4
3	Conclusion.....	4

1 Introduction

Dans le cadre de ce projet, nommé Concer'ino, nous devons réaliser une application complète de base de données relationnelle. Ce projet est à réaliser en groupe de trois personnes. Notre projet consistera en une application permettant de lister différents concerts dans toute la suisse avec la possibilité d'indiquer sa présence.

Le présent document définit la modélisation conceptuelle dudit projet.

2 Modélisation conceptuelle

2.1 Choix de modélisation

Le login d'un utilisateur correspond à son adresse email.

Afin de permettre à un artiste de faire à la fois une carrière solo et de faire partie d'un groupe, nous avons défini une entité parente « Artiste » qui a comme enfants « ArtisteSolo » et « Groupe ». Ceci nous permet de facilement savoir si un artiste correspond à une personne ou un groupe. De plus, pour pouvoir gérer l'historique des groupes nous avons des attributs d'association entre « ArtisteSolo » et « Groupe », nous permettant de savoir quand un artiste a rejoint un groupe et l'a quitté.

Nous avons décidé de ne pas créer une super-entité « Note » pour regrouper les 3 types de notes car nous n'utiliserons jamais lesdites données en même temps.

Concernant les attributs d'association :

- « note » correspond à un entier de 0 à 5 compris
- « date » est généré automatiquement et correspond à la date du jour.

Pour pouvoir gérer la suppression des concerts par leur créateur, il est nécessaire de connaître ce dernier, ceci nous force donc à avoir une relation supplémentaire entre les tables « Utilisateur » et « Concert ».

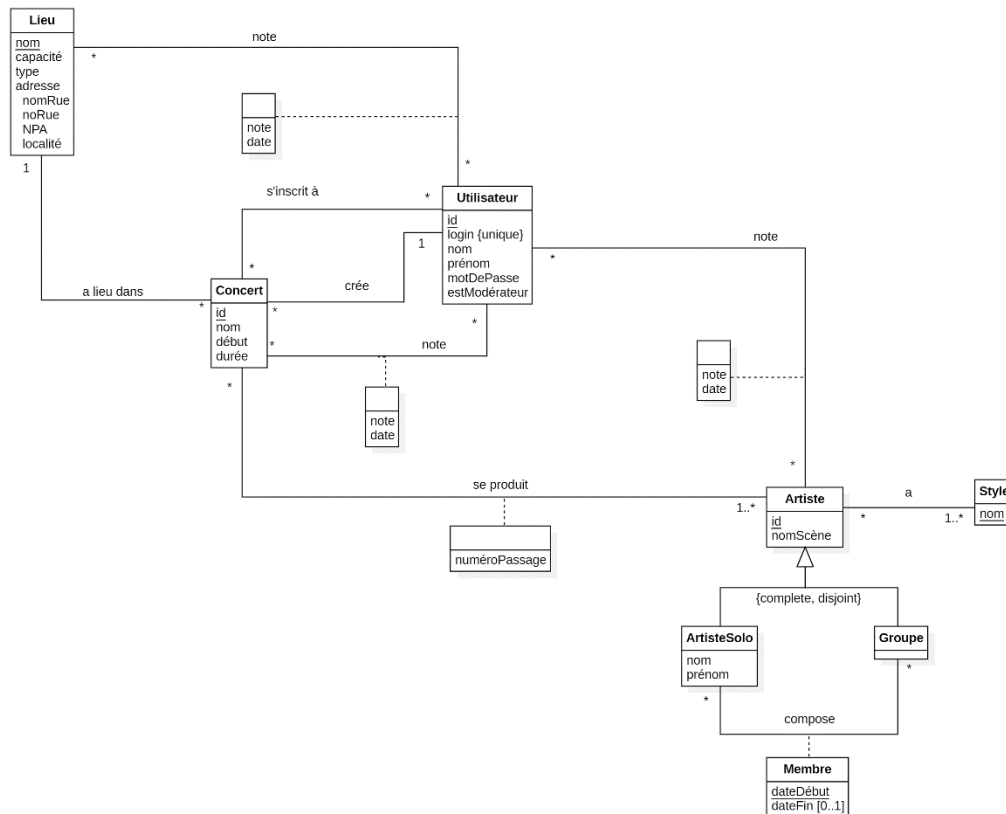
L'attribut « début » d'un concert correspond à un timestamp définissant sa date et son heure de début.

L'attribut « type » d'un lieu est un type énuméré pouvant prendre une valeur de l'ensemble {Intérieur, Extérieur}.

2.2 Contraintes d'intégrité

- Plusieurs concerts ne peuvent pas avoir lieu en même temps et au même endroit.
- Le nombre de personne inscrite à un concert ne peut pas excéder la capacité de sa salle.
- Un artiste ne peut pas se produire lors de plusieurs concerts en même temps.
- Un utilisateur ne peut pas s'inscrire à différents concerts ayant lieu en même temps.
- Pour un même concert, 2 artistes ne peuvent pas avoir le même numéro de passage.
- Le numéro de passage est une suite (commençant à 1) continue de valeurs.
- Un concert ne peut pas être créé avec une date de début antérieure à la date courante.
- Un utilisateur ne peut pas s'inscrire à un concert dont la date est déjà passée.
- Un concert ne peut être noté que par un utilisateur ayant assisté audit concert.
- Une salle ne peut être notée que par un utilisateur ayant assisté à au moins un concert dans cette dernière.
- Un artiste ne peut être noté que par un utilisateur ayant effectivement assisté à une de ses performances.
- Un groupe sans membres actifs ne peut pas se produire en concert.
- La date de début d'un membre doit être antérieure ou égale à la date de fin.
- Un artiste solo ne peut pas rejoindre un groupe dont il fait déjà partie.

2.3 Schéma



3 Conclusion

Nous avons eu quelques problèmes pour la modélisation des notes que l'utilisateur peut mettre sur les groupes, les artistes et les salles. Nous sommes d'abord partis sur une modélisation à l'aide d'un héritage plutôt que la solution actuelle.

Un autre problème que nous avons rencontré est celui des artistes solos. Nous n'avions pas prévu de gérer ce cas-là car nous pensions qu'il serait beaucoup trop complexe à gérer. Avec la solution que nous utilisons, nous devrions être capable de gérer cette partie aisément.

Globalement, nous pensons qu'avec notre modélisation actuelle, la suite du projet devrait se dérouler sans difficultés majeures.