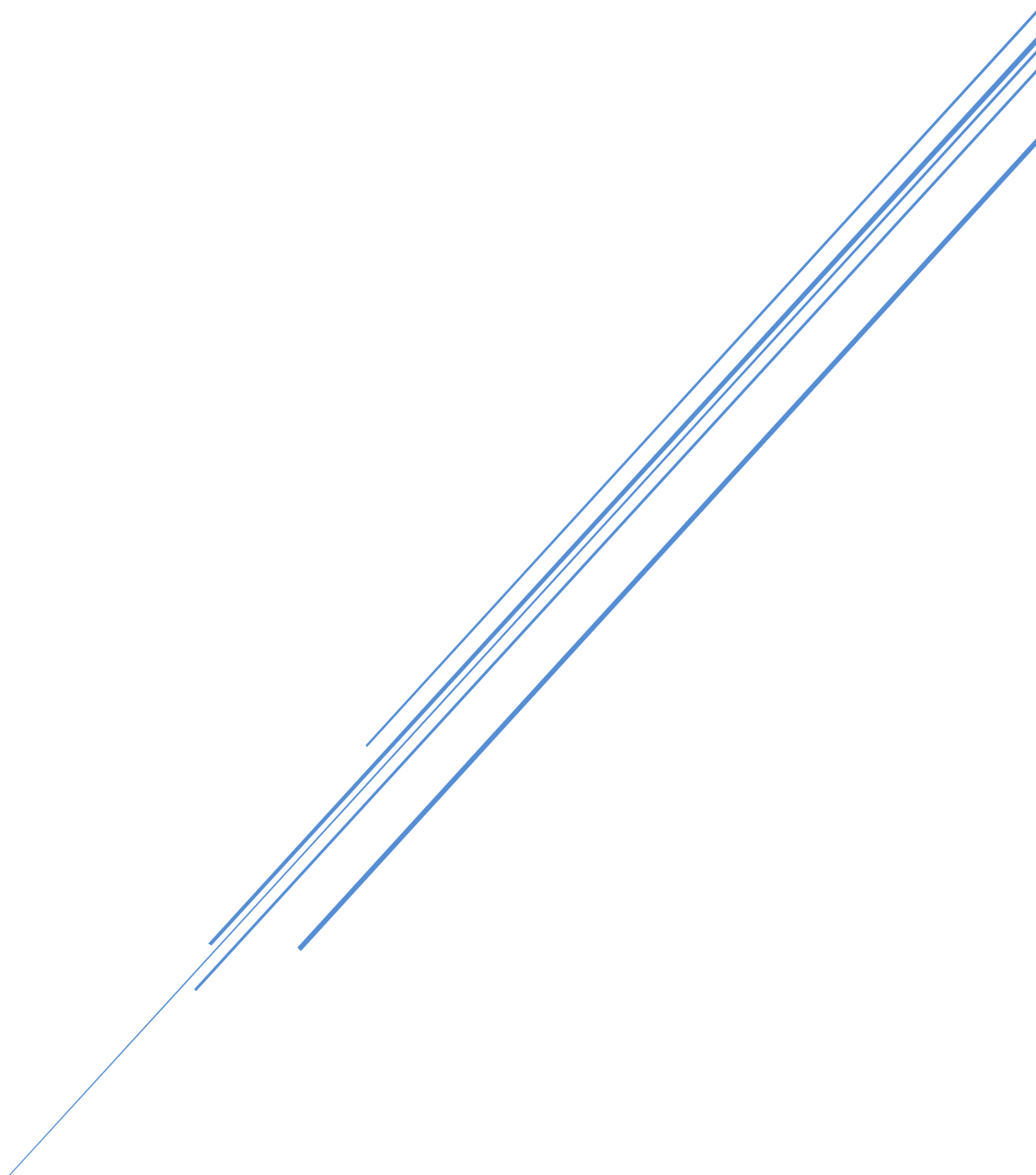


PROJET CONCERT'INO

Groupe 4 – Marengo – Friedli - Marzullo



BDR

Table des matières

1	Introduction	2
2	Descriptif du projet	2
3	Modèle EA	3
4	Modèle relationnel	4
4.1	Représentation graphique	4
4.2	Description des tables / attributs	5
4.3	Cohérence des données	6
5	Description de l'application	6
6	Bugs connus	7
7	Conclusion	7
8	Annexes	7

1 Introduction

Le but du projet Concer'ino est de réaliser une application complète de base de données relationnelle. Cette application permet de lister différents concerts dans toute la Suisse avec la possibilité d'indiquer sa présence. Ce projet a été réalisé dans le cadre du cours BDR, par Jonathan Friedli, Stéphane Marengo et Loris Marzullo.

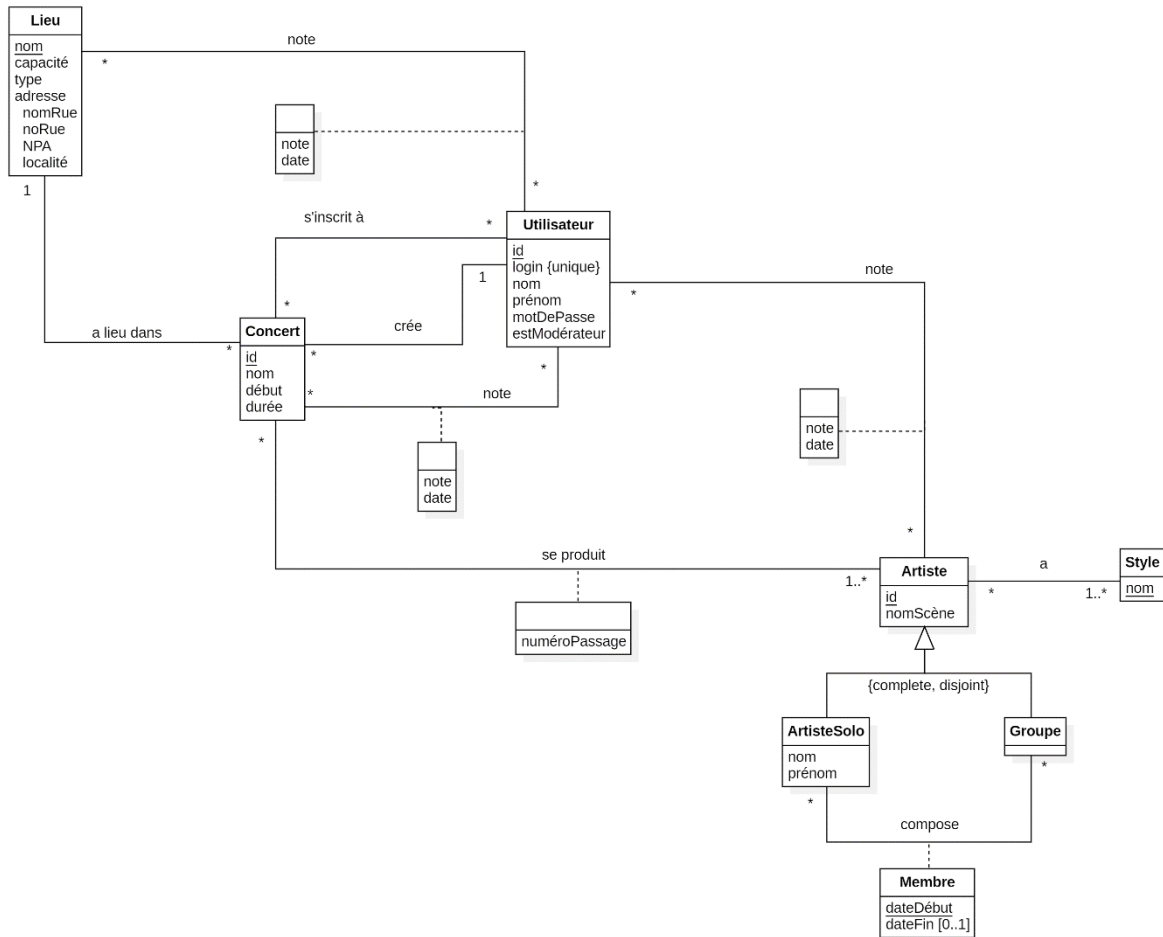
2 Descriptif du projet

Les données que nous traitons sont principalement les concerts, mais aussi les lieux où ceux-ci se déroulent et les différents artistes qui s'y produisent. L'application est divisée en différentes pages afin de visualiser ces informations séparément.

Pour modifier la base de données, un utilisateur peut se créer un compte et se connecter avec son adresse électronique. Lorsqu'il consultera les concerts à nouveau, il pourra alors décider de s'y inscrire. Il aura également accès à différents formulaires permettant d'ajouter des concerts, des lieux, des artistes et des groupes. Il aura enfin la possibilité de les évaluer en leur affectant une note.

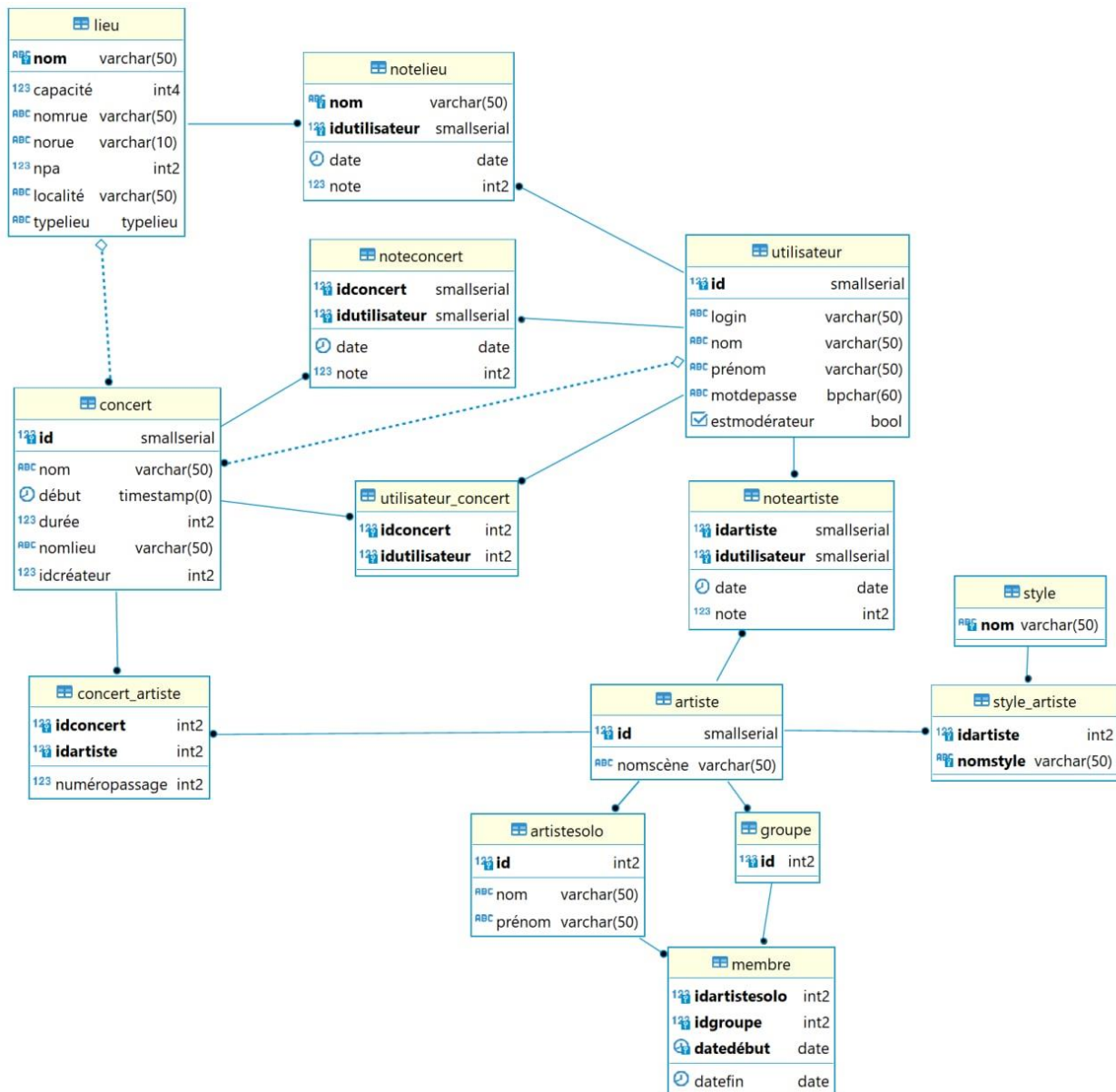
De plus, des utilisateurs peuvent être promus modérateurs et obtenir des privilèges comme assigner le rôle modérateur à d'autres utilisateurs et supprimer des concerts. Cette dernière option est aussi disponible pour un utilisateur standard à condition qu'il soit le créateur du concert en question.

3 Modèle EA



4 Modèle relationnel

4.1 Représentation graphique



4.2 Description des tables / attributs

Un concert est défini par un nom, une heure de début, une durée, un lieu et un utilisateur qui en est le créateur. La table *Utilisateur_Concert* fait le lien entre les utilisateurs et les concerts auxquels ils se sont inscrits.

Un lieu représente l'endroit où se déroule un concert et est défini par une capacité, une adresse (nom et numéro de rue, NPA et localité) ainsi que d'un type de lieu. Celui-ci est un *enum TypeLieu* pouvant valoir « Intérieur » ou « Extérieur ».

Un artiste correspond à une performance qui a lieu durant un concert, cela peut être un artiste solo ou un groupe. Dans les deux cas, un artiste est défini par un nom de scène. Les artistes solos possèdent également un nom et un prénom.

Un groupe est composé de différents artistes solos, appelés membres. Cette table contient la date à laquelle ils ont rejoint ledit groupe et, éventuellement, la date à laquelle ils l'ont quitté.

Chaque artiste joue au moins un style de musique, la table *Style_Artiste* fait le lien entre ceux-ci.

La table *Concert_Artiste* indique les concerts où performant les artistes. Un numéro de passage distinct est attribué à chacun d'eux afin d'établir leur ordre de passage.

Un utilisateur est défini par un login (sous forme d'adresse électronique), un nom, un prénom et un mot de passé (hashé sur 60 chars avec l'algorithme *Bcrypt*). Il peut également être promu modérateur afin d'obtenir des droits supplémentaires.

Un utilisateur peut attribuer des notes aux concerts, artistes et lieux. La note, qui est un entier de 0 à 5, ainsi que la date à laquelle celle-ci a été affectée sont stockées dans une des tables *NoteConcert*, *NoteArtiste* et *NoteLieu*, selon ce qui a été noté.

4.3 Cohérence des données

Il n'y a aucune validation des données sur l'entier de l'application. Les différentes contraintes d'intégrités de notre base de données sont gérées uniquement à l'aide de triggers et de procédures stockés.

La cohérence des données est garantie, à l'exception de l'ordre de passage des artistes. En effet, nous avons un trigger qui se déclenche après chaque mise à jour dans cette table et vérifie que la valeur modifiée ou insérée corresponde bien à valeur finale de la suite courante.

Ceci rend la modification de numéro de passage impossible même lorsque deux numéros sont intervertis dans la même transaction car le trigger ne dépend pas de celle-ci. Pour faire les choses correctement il aurait fallu que ce trigger soit déclenché à la fin d'une transaction et non pas à chaque modification d'enregistrement.

A cause de ce comportement nous n'avons pas été en mesure de mettre en place un système permettant de mettre à jour automatiquement les numéros de passage lors d'une suppression d'un enregistrement de la table *Concert_Artiste*.

5 Description de l'application

La page d'accueil de l'application affiche les 5 prochains concerts qui auront lieu. Il est ensuite possible de naviguer entre les différentes pages grâce au menu situé sur la partie supérieure de ladite page.

Chacune de ces pages liste différentes données et il est alors possible de cliquer sur une d'entre elles afin d'accéder à une page de détails. De plus, si un utilisateur est connecté, il aura accès à un lien le menant à un formulaire dans le but d'ajouter des données (sauf pour les utilisateurs).

La page de détails d'un concert permet de connaître les artistes qui s'y produisent et les utilisateurs qui s'y sont inscrits. Pour un utilisateur connecté, il peut s'y inscrire ou s'y désinscrire et supprimer le concert s'il en a les droits.

La page de détails d'un artiste indique les groupes auquel il fait ou a fait partie et la page d'un groupe liste les artistes le constituant. Au bas de ces pages, une liste d'artistes ayant au moins un style en commun avec l'artiste consulté est proposée.

Dans toutes ces pages de détails, ainsi que dans celle des lieux, la moyenne des notes totales est calculée et un champ est disponible afin d'entrer une nouvelle note.

La page de détails des utilisateurs affiche les notes qu'il a attribué, les concerts qu'il a créés, ainsi que les concerts auxquels il a participé et auxquels s'est inscrit.

6 Bugs connus

- Si un utilisateur connecté accède à une page qui n'existe pas, le menu s'affiche comme s'il n'était pas connecté.

7 Conclusion

Finalement, l'application est fonctionnelle et le projet a pu être terminé. Cependant, certains points pourraient être améliorés :

- L'interface utilisateur n'est pas très pratique lors des ajouts de données.
- En cas d'erreur, il n'y a pas de message détaillant la source de l'erreur en question.
- Lors de la suppression d'un artiste à un concert, la cohérence des numéros de passage n'est pas garantie comme décrit dans la section 4.3.
- Lorsqu'un utilisateur souhaite créer un concert, la liste de tous les artistes est disponible. Il faudrait plutôt autoriser uniquement les artistes disponibles en fonction de la date et l'heure du concert.
- Pour l'instant, un artiste ne peut rejoindre qu'un groupe et ne peut jamais le quitter.
- Il n'est pas possible d'éditer les différentes données que l'application nous permet d'ajouter.

Dans l'ensemble nous sommes plutôt contents du résultat final même si l'interface laisse à désirer. La base de données quant à elle est, selon nous, plutôt bien gérée ce qui est le point principal du projet et le point sur lequel nous avons le plus travaillé.

8 Annexes

L'adresse suivante <https://github.com/Marinlestyle/Concerino> permet de consulter :

- Le code source du projet
- Le guide d'installation
- Les anciens documents liés au projet