

Contents

Notes sur le SQL	2
proj_TypeUtilisateur	2
TypeUtilisateur	2
proj_Utilisateur	3
IDU	3
login	3
passHash	3
nom	3
prenom	3
adressePostale1	3
adressePostale2	3
codePostal	4
ville	4
adresseMail	4
telephone	4
<i>typeUtilisateur</i>	4
proj_Spectacle	4
idSpectacle	4
nom	5
type	5
infos	5
proj_Salle	5
idSalle	5
nom	5
adresse	6
type	6
proj_Representation	6
idRepresentation	6
<i>idSalle</i>	6
<i>idSpectacle</i>	6
date	6
HoraireDebut	6
HoraireFin	7
proj_Categorie	7
idCategorie	7
Categorie	7
<i>idSalle</i>	7
proj_Place	7
idPlace	7
<i>idSalle</i>	7
<i>idCategorie</i>	8
Handicap	8
Rang	8
Numero	8

proj_PrixPlace	8
<i>idCategorie</i>	8
<i>idSpectacle</i>	8
prix	8
proj_Reservation	9
<i>IDU</i>	9
<i>idRepresentation</i>	9
<i>idPlace</i>	9
messages_contacts	9
id_message_contact	9
Nom_contact	9
Prenom_contact	9
Mail_contact	10
Text_message	10
date_message_contact	10
Disponibilité des places	10
Clés étrangères	10

Notes sur le SQL

Notes table par table

Toutes les tables commencent par le préfix “proj_”

proj_TypeUtilisateur

TypeUtilisateur

Défini les types d'utilisateurs possibles:

- admin : administre le site et la base de données
- manager : peut gérer les spectacles et les réservations mais pas le site et la base
- user : client du site, peut créer un compte et faire des réservations

Un utilisateur du site doit obligatoirement avoir un TypeUtilisateur, seul un admin peut changer le type d'un utilisateur.

varchar 10

proj_Utilisateur

IDU

Identifiant Unique utilisateur, **clé primaire**

type numérique auto-incrémenté

login

Identifiant de l'utilisateur (pseudo, adresse mail)

Varchar 30, **UNIQUE**

Il ne doit pas y avoir 2 login identiques dans la base

passHash

Mot de passe haché par la fonction password_hash() de PHP; doit être utilisé directement par password_verify().

varchar 255 NOT NULL

nom

Nom de l'utilisateur

Varchar 30

prenom

Prenom de l'utilisateur

Varchar 30

adressePostale1

Première ligne de l'adresse postale

Varchar 50

adressePostale2

Deuxième ligne de l'adresse postale

Varchar 50

codePostal

Code postal : traité comme string pour les client étrangers

Varchar 10

ville

Ville de résidence

Varchar 15

adresseMail

email de contact, *devra correspondre à un certain format*

unique

varchar 50

telephone

telephone de contact, traité comme string pour les numéros étrangers, *devra correspondre à un certain format*

varchar 15

typeUtilisateur

référence le type d'utilisateur

Par défaut, réglé sur *user*, devra être changé ultérieurement par un admin

Clé étrangère sur **proj_TypeUtilisateur.TypeUtilisateur**

varchar 10

proj__Spectacle

Référence tous les spectacles proposés (faits, en cours, à venir)

idSpectacle

Identifiant unique auto-incrémenté, **clé primaire**

Int

nom

Nom du spectacle

varchar 50

(unique ?)

type

Référence le type de spectacle: concert, pièce de théâtre, danse ...

Suggestion: Les valeurs devraient venir d'une table externe

varchar 30

infos

contient les informations du spectacle:

- nom du metteur en scène / chorégraphe
- noms des acteurs/musiciens
- description courte

suggestion : peut être divisé en plusieurs champs dans un vrai site

TEXT

proj__Salle

Une salle est composée de places et appartient à un lieu

idSalle

Identifiant unique

Numérique INT auto-incrémenté **clé primaire**

nom

nom de la salle (salle n°3; Théâtre Odéon; etc)

varchar 30

adresse

adresse postale

TEXT

type

type de salle: salle de concert, théâtre, piste de cirque. . .

varchar 20

proj_Representation

Une représentation est identifiée par un spectacle (auteur, acteurs, etc), une salle, un horaire et une date. Un spectacle peut avoir plusieurs représentations dans la même salle.

idRepresentation

Identifiant unique

Numérique INT auto-incrémenté **clé primaire**

idSalle

proj_Salle.idSalle, **clé étrangère**

idSpectacle

proj_Spectacle.idSpectacle, **clé étrangère**

date

date de la représentation, séparée du temps pour être plus simple à dupliquer

DATE

HoraireDebut

Début de la représentation

TIME

HoraireFin

Fin de la représentation

TIME

proj_Categorie

Catégories disponibles pour une salle

idCategorie

Identifiant unique

Numérique INT auto-incrémenté **clé primaire**

Categorie

Nom de la categorie

varchar 20

idSalle

Les salles ont certaines catégories mais pas toutes. Il faut séparer les catégories de chaque salle, car la même catégorie n'aura pas le même prix dans un théâtre que dans un opéra.

proj_Salle.idSalle, **clé étrangère**

proj_Place

idPlace

Identifiant unique

Numérique INT auto-incrémenté **clé primaire**

idSalle

Une place appartient toujours à une salle

proj_Salle.idSalle, **clé étrangère**

idCategorie

proj_Categorie.idCategorie, **clé étrangère**

Handicap

Détermine si la place est accessible ou non.

varchar (3)

Rang

Rangée à laquelle la place appartient

varchar (5)

Numero

Numéro de la place: concaténation Rang + idPlace

varchar(10)

proj_PrixPlace

Le prix est fonction du spectacle, peut importe la représentation, et de la catégorie. La catégorie permet de remonter à la salle.

idCategorie

proj_Categorie.idCategorie, **clé étrangère; clé primaire**

idSpectacle

proj_Spectacle.idSpectacle, **clé étrangère; clé primaire**

prix

Prix en euros

Décimal

proj__Reservation

Une réservation prend en compte 1 et 1 seule place, pour une unique représentation, par un unique utilisateur /client.

Avec le couple idPlace et idRepresentation, il est possible de retrouver le prix

IDU

proj__Utilisateur.IDU, **clé étrangère; clé primaire**

idRepresentation

proj__Representation.idRepresentation, **clé étrangère; clé primaire**

idPlace

proj__Place.idPlace, **clé étrangère; clé primaire**

messages__contacts

Cette table est destinée à contenir les messages envoyés à travers la page infos.phpMyAdmin

id_message_contact

Id unique du message, généré à l'insertion

type *SERIAL*

Nom_contact

Nom de l'expéditeur

varchar 30

Prenom_contact

Prenom de l'expéditeur

varchar 255

Mail_contact

email de l'expéditeur

varchar 255

Text_message

Texte du message

type *TEXT*

date_message_contact

Date d'enregistrement du message

type *date*

Disponibilité des places

Pas besoin de créer une liste spéciale:

Il suffit de trier les places pour le spectacle concerné où il y a (ou pas) de réservations de faite.

Clés étrangères

proj_Utilisateur(typeUtilisateur) -> proj_TypeUtilisateur(typeUtilisateur)

proj_Representation(idSalle) -> proj_Salle(idSalle)

proj_Representation(idSpectacle) -> proj_Spectacle(idSpectacle)

proj_Categorie(idSalle) -> proj_Salle(idSalle)

proj_Place(idSalle) -> proj_Salle(idSalle)

proj_Place(idCategorie) -> proj_Categorie(idCategorie)

proj_PrixPlace(idCategorie) -> proj_Categorie(idCategorie)

proj_PrixPlace(idSpectacle) -> proj_Spectacle(idSpectacle)

proj_Reservation(IDU) -> proj_Utilisateur(IDU)

proj_Reservation(idRepresentation) -> proj_Representation(idRepresentation)

proj_Reservation(idPlace) -> proj_Place(idPlace)