**Projet Bases de données avancées**

Participants :

TRAORE Paul

CADIOU Léo-Paul

Description générale du projet :

Nous avons décidé d’orienter notre projet semi-libre de BDA vers un système de gestion de festival.

Notre système permettra de se renseigner sur des festivals , consulter les éditions de ces derniers et obtenir des informations sur les films présentés ainsi que les lieux où se présenterons les éditions consultées.

Il sera aussi possible à un utilisateur ( possédant un compte ) de se renseigner sur les places disponibles au festival de son choix et d’en acheter selon diverses catégories de places.

Nous aurons aussi une partie concernant les films , s’ils sont présentés ou non dans un festival , ainsi qu’un descriptif contenant notamment les personnes ayant travaillé sur ce même film ou même une critique sur le film.

L’utilisateur pourra également consulter les personnes invitées aux différents festivals ( par exemple , les acteurs d’un film, le producteur ).

Cette base de données pourra par exemple être utilisée dans le cadre d’une application de réservation/consultation de place de festival ainsi que les films proposés à ce dernier.

Modélisation :

Nous avons décidé d’organiser notre BDD autour d’une table “**Edition**” , qui permet de définir une édition donnée d’un festival ( ex : 3eme édition du festival de Canne à sucre ) , correspondant à un lieu , une année , date de début , et une date de fin .

De cette table nous avons des liaisons vers différentes table , à savoir :

* “**Festival**” , permet de représenter un festival en lui même , elle définit le thème , le nom , la date de création et d’autre éléments qui décrivent un festival.
* “**Personne\_Invitée**” , qui fera la liaison entre les personnes ayant travaillé sur un film et leurs présence à une Édition d’un festival . Ces personnes seront définis par leurs Id\_personne , le film sur lequel ils ont travaillé , l'édition à laquelle ils se rendent , ainsi que leurs métier sur le tournage du film.
* “**Film\_Edition**” , permettra de faire la liaisons entre une Édition et les films présentés à cette dernière. ( pour une éditions donnée ‘x’ et un film ‘y’, la présence du couple (x,y) dans la table implique la présence du film ‘y’ à l'Édition ‘x’ ). Par la même occasion , nous avons définis la table “**Film**” :
  + Un film sera composé d’un nom de film , son type ( le thème principal du film ) , la durée ( en minutes ) et la date de parution de ce dernier. A ce film seront relié des “**Critique**” qui seront définie par une note , un avis générale ainsi que le “**Site**” ayant publié cette même critique , lui même définis par un nom et un lien.
* “**Catégorie**” , contient les catégories de places disponibles pour une éditions donnée, ainsi que le prix , la capacité maximale pour la catégorie choisie ( en nombre de place ) , ainsi qu’un descriptif ( ex : “Catégorie 1” / “VIP”) . De la table “**Catégorie**” , nous disposons d’une table “**Places**” .

Cette table , qui nous permet d’associer une place ( réservée ) avec un Id , et un numéro à un Utilisateur ( possédant un compte ) , cette dernière peut être au nom de l’utilisateur ou au nom d’une autre personne ( chaque place sera nominative ) .

* “**Localisation**” , associe deux coordonnées GPS ( à savoir longitude, latitude ) à un “**Lieu**” pour déterminer le lieu où se déroulera l'Édition concernée, la table “**Lieu**” sera défini comme tel :
  + Un département , une ville , un pays , une adresse , un code postale.

Nous avons également une table “**Personne**” , définis par un nom , prénom et une date de naissance. De cette dernière sont issues d’autres tables , à savoir :

* “**Utilisateur**” , qui correspond à un utilisateur de notre système , ce dernier étant associé à un Email , un mot\_de\_passe , un numéro de téléphone , une “**Localisation**” . Ces derniers serviront au moment de l’achat d’une place . Par la même occasion, l'utilisateur dispose d’un “**Compte**” ( qui correspondra à un porte-monnaie virtuel ), qui comportera un solde et un découvert autorisé .
* “**Personne\_Invitée**” , qui correspond aux personnes invitées à une Édition de festival donnée ( associé à un film et un rôle ) , cette dernière est définie plus haut.
* “**Participant**” , qui représente le “casting” d’un film , sera composé d’une personne , d’un film , et d’un rôle , le tout pour permettre d’obtenir une clé primaire Unique “Id\_Participant” ( a savoir que nous avons permis à une même personne , de remplir plusieurs “rôle” dans un même film ).

**Fonctionnalités Globales autour de la Base de Données:**

**Autour de l’utilisateur :**

\_ Un utilisateur peut acheter une ( ou plusieurs ) place à son nom ( ou pour une autre personne ) pour une édition d’un festival.

\_ Un utilisateur ne peut pas dépenser plus que le découvert autorisé de son compte.

\_ Un utilisateur peut consulter un registre contenant un ensemble de festival.

( et donc obtenir toutes les informations qu’il souhaite sur ce dernier )

\_ Un utilisateur peut consulter un registre contenant un ensemble de film.

\_ Un utilisateur peut accéder à la liste des personnes ayant participé à un film.

\_ Un utilisateur peut consulter une liste de film liés à une édition d’un festival.

\_ Un utilisateur peut se désinscrire de la BD sous certaines condition:

\_ il n’a pas acheté de place pour un festival prévu à une date ultérieur

\_ son compte est supprimable (son solde doit être à 0)

**Autour d’un compte :**

\_ Un compte ne peut pas avoir un solde inférieur à son découvert autorisé

\_ Si le solde du compte est inférieur au prix moyen d’une place, l’utilisateur doit être averti.

\_ Un compte ne peut pas être fermé si son solde est différent de 0.

\_ Un compte ne peut pas être supprimé si on ne supprime en même temps que l’utilisateur.

**Autour d’un festival et d’une édition :**

Un festival peut avoir lieu sur une année à diverses endroits mais cela implique que ce festival pour l’année en question soit lié à plusieurs instances “**Édition**” et que les dates ne se chevauchent pas.

\_ On peut mettre à jour la liste de film présenté à une édition de festival, cependant celle-ci doit contenir au minimum 1 film.

\_On peut réserver un place à une édition de festival uniquement si la capacité maximale du lieu n’a pas été atteinte.

\_On peut réserver un place à une édition de festival uniquement si la capacité maximale de la catégorie de place en question n’a pas été atteinte.

\_ On peut définir différentes catégories de place.

\_ On peut connaître le nombre d’éditions d’un festival au fil du temps en fonction de sa périodicité.

\_ On peut répertorier les différentes personnes ayant participé à un festival.

\_ On peut lister tous les films présenté à un festival depuis la date de création du festival.

\_ On peut consulter la note globale de tout les film présentés lors d’une ou l’ensemble des éditions d’un festival.

\_ On peut mettre à jour le nombre de catégorie pour une édition.

**Autour d’une Place :**

\_ On peut affecter une certaine catégorie et donc un prix à une place.

\_ On peut réserver une place avec son compte pour une autre personne.

\_ On peut demander un remboursement d’une place si l'événement n’a pas encore eu lieu

**Autour d’une Catégorie :**

**\_** On peut calculer la catégorie la plus populaire.

\_ On peut savoir quelle est le prix moyen des places proposés par une ou toute les éditions de festival.

\_ On peut calculer le nombre de places restantes dans une catégorie donnée à une édition de festival souhaitée.

\_ On mettre à jour le nombre de

**Autour d’un Film et de ses Participants:**

\_ On peut mettre à jour les Participants d’un film.

\_ On peut lister tous les festivals dans lesquels le film a été présenté.

**Autour des Critiques de film et des Sites de Critiques :**

\_ On peut mettre à jour la note globale et l’avis générale par une site de critique sur un certain film.

\_ La note fournis par le site doit fluctuer entre 0 et 10.

**Autour du Lieu et de la Localisation :**

\_ On peut faire statistique sur la répartition des festivals sur l’ensemble d’un territoire.

\_ On peut faire statistique sur la répartition des utilisateurs sur l’ensemble d’un territoire.

**Triggers :**

(Les fonctions secondaires ne seront pas nécessairement implémentées si l’on peut s’en passer)

\_ Lorsqu’un utilisateur tente de se désinscrire mais qu’une place en son nom est réservé pour une édition de festival ayant lieu à une date ultérieur.

Il faut aussi vérifier que le solde du compte est différent de 0.

CREATE TRIGGER suppression

CREATE OR REPLACE suppression\_utilisateur() RETURNS TRIGGER

CREATE OR REPLACE suppression\_compte(Id\_compte INTEGER)

RETURNS BOOLEAN

\_ Lorsqu’on supprime un film\_edition mais que ce film est le seul présenté lors de cette édition de festival alors un message d’erreur survient.

CREATE TRIGGER suppression\_film\_edition

CREATE OR REPLACE est\_supprimable(suppr\_edition BOOLEAN)

RETURNS TRIGGER

(renvoie le trigger si on ne veut pas supprimer d’edition mais juste un film\_edition)

\_ Lorsqu’on supprime une édition de festival, il faut supprimer aussi tous les films qui devaient être présentés lors de cette édition de festival.

CREATE TRIGGER suppresssion\_edition

CREATE OR REPLACE suppression\_edition() RETURNS TRIGGER

(utilisation de la fonction est\_supprimable(true) )

\_ Lorsqu’un utilisateur souhaite acheter une place pour une édition d’un festival mais qu’il n’y a plus de place dans la catégorie de place demandée ou plus de place du tout pour l'événement en question alors un message d’erreur survient pour lui indiquer la raison du refus.

CREATE TRIGGER achat()

CREATE OR REPLACE place\_dispo\_categorie() RETURNS TRIGGER

CREATE OR REPLACE place\_dispo(Id\_edition INTEGER) RETURNS BOOLEAN

\_ Lorsqu’on supprime une catégorie de place pour une édition de festival mais que la catégorie était la seule et unique catégorie de place pour cet évènement.

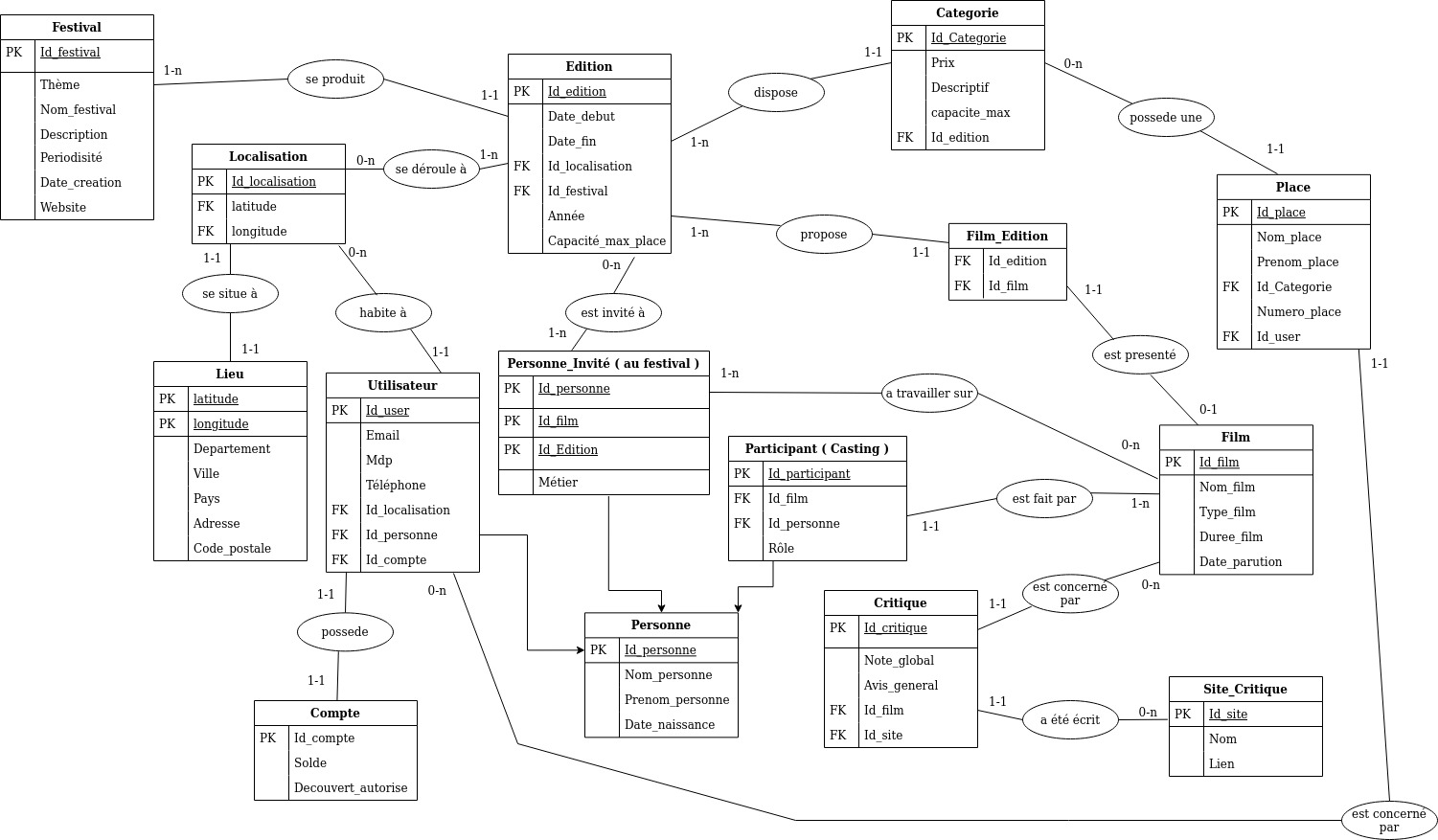
CREATE TRIGGER suppression\_categorie

CREATE OR REPLACE categorie\_supprimable() RETURNS TRIGGER

Tableau des contraintes :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | **Attribut** | **Contrainte** | **Résolution** | **Commentaire** |
| Personne | prenom\_personne | !=vide | NOT NULL | le prénom d’une personne doit être complété |
|  | nom\_personne | != vide | NOT NULL | le nom d’une personne doit être complété |
|  | date\_naissance | personne a + de 18 ans | CHECK | plus simple qu’un trigger |
|  | nom\_personne, prenom\_personne,  date\_naissance | une personne est unique | UNIQUE | unicité de 3-uplet d'attribut représentant l’identité d’un personne |
|  |  |  |  |  |
| Personne Invitée | métier | le métier doit être correct | CHECK | le métier sélectionné doit être dans un liste de métier possible |
|  |  |  |  |  |
| Film | Nom\_film | ne peut pas être vide | NOT NULL | un film possède obligatoirement un nom |
|  | Type\_film | doit être un type correct | CHECK | le type de film doit être dans une liste possible de type |
|  | Duree\_film | le durée ne peut être vide | CHECK | vérifier si la durée est supérieur à 0 minute |
|  |  |  |  |  |
| Participant | Rôle | doit être un rôle valide | CHECK | vérifie si le rôle est un rôle possible dans un film |
|  | Id\_film, Id\_personne, Role | un participant avec un rôle précis est unique | UNIQUE | une personne , pour un film et un rôle donnés , constitue un participant unique |
|  |  |  |  |  |
| Utilisateur | Email | ne doit pas être vide | NOT NULL | Une adresse mail est nécessaire |
|  | Mdp | ne doit pas être vide | NOT NULL | Un mdp est forcément nécessaire |
|  | Email,Mdp | le couple doit être unique | UNIQUE | Un email et un MDP donnés font référence à un seul utilisateur |
|  | Id\_personne | un utilisateur correspond à une personne | UNIQUE | car il a une seul personne qui correspond à un seul utilisateur |
|  | Id\_compte | un personne ne possède qu’un seul compte | UNIQUE | un compte n’est attribué qu'à une seul personne |
|  |  |  |  |  |
| Compte | Solde | toujours supérieur au découvert autorisé | CHECK | Permet de vérifier si le solde est bien supérieur au découvert |
|  | Solde | jamais vide | NOT NULL | le solde doit etre superieur a zero et contenir une certaine somme ( même 0€ ) |
|  | Decouvert\_autorise | jamais vide | NOT NULL | le découvert doit être renseigné |
|  |  |  |  |  |
| Lieu | Departement | jamais vide | NOT NULL | nécessaire pour l’identification d’un lieu |
|  | Pays | jamais vide | NOT NULL | nécessaire pour l’identification d’un lieu |
|  | Code\_postal | jamais vide | NOT NULL | nécessaire pour l’identification d’un lieu |
|  | Ville | jamais vide | NOT NULL | nécessaire pour l’identification d’un lieu |
|  |  |  |  |  |
| Festival | Nom\_festival | jamais vide | NOT NULL | nécessaire pour l’identification d’un festival |
|  | Nom\_festival | deux festivals ne peuvent avoir le même nom | UNIQUE | Un nom de festival fait référence à un seul et unique festival |
|  | Website | un site web est unique | UNIQUE | un site web ne peut faire référence à plusieurs festivals |
|  |  |  |  |  |
| Edition | Id\_festival, Annee, Date\_debut, Date\_fin | un festival ne peut pas avoir lieu dans une même année aux mêmes dates | UNIQUE | un festival à une date donnée ne peut se produire qu’une seule fois |
|  |  |  |  |  |
| Place | Nom\_place | jamais vide car place nominative | NOT NULL | Les places sont forcément composée d’un nom |
|  | Prenom\_place | jamais vide car place nominative | NOT NULL | Les places sont forcément composée d’un prenom |
|  | Id\_festival, Nom\_place, Prenom\_place | les places sont nominatives pour chaque festival | UNIQUE | Une personne ne peut pas avoir plusieurs place à son nom/prénom sur le même festival |
|  | Id\_festival, Numero\_place | place avec numéro unique pour un festival | UNIQUE | les places d’un festival sont numérotés de 1 à n indépendamment des catégories |
|  |  |  |  |  |
| Site\_Critique | Nom | jamais vide | NOT NULL | un site comporte un nom |
|  | Lien | jamais vide | NOT NULL | un site est composé d’un lien |
|  | Lien | un lien par site | UNIQUE | un lien fait référence à un seul site |
|  |  |  |  |  |
| Critique | Note\_globale | jamais vide | NOT NULL | jamais nul car définie une critique de film |
|  | Note\_globale | notation comprise en 0 et 10 | CHECK | On vérifie que la note est bien comprise entre 0 et 10 |
|  | Id\_film, Id\_site | un site note une seule fois un film | UNIQUE | une critique provient d’un site pour un film |

Digramme :



Create\_table :

CREATE TABLE Personne(

Id\_personne INTEGER PRIMARY KEY,

nom\_personne VARCHAR(50) NOT NULL,

prenom\_personne VARCHAR(50) NOT NULL,

date\_naissance DATE CHECK (date\_naissance < now()- interval'18 year' ),

UNIQUE(nom\_personne,prenom\_personne,date\_naissance)

);

CREATE TABLE Personne\_Invitee(

Id\_personne INTEGER,

Id\_film INTEGER,

Id\_festival INTEGER,

metier VARCHAR(50) CHECK(metier IN ('Acteur','Réalisateur','Producteur','Cameraman')) NOT NULL,

PRIMARY KEY(Id\_personne, Id\_film, Id\_festival),

FOREIGN KEY(Id\_personne) REFERENCES Personne(Id\_personne),

FOREIGN KEY(Id\_film) REFERENCES Film(Id\_film),

FOREIGN KEY(Id\_edition) REFERENCES Edition(Id\_edition)

);

CREATE TABLE Film(

Id\_film INTEGER PRIMARY KEY,

Nom\_film VARCHAR(50) NOT NULL,

Type\_film VARCHAR(50) CHECK( type\_film IN ('Documentaire', 'Action','Drame','Animation')) ,

Duree\_film INTEGER CHECK( duree\_film > 0),

Date\_parution DATE

);

CREATE TABLE Film\_Edition(

Id\_edition INTEGER,

Id\_film INTEGER,

PRIMARY KEY(Id\_edition, Id\_film),

FOREIGN KEY(Id\_film) REFERENCES Film(Id\_film),

FOREIGN KEY(Id\_edition) REFERENCES Edition(Id\_edition)

);

CREATE TABLE Participant(

Id\_participant INTEGER PRIMARY KEY,

Id\_film INTEGER,

Id\_personne INTEGER,

Role VARCHAR(50) CHECK(metier IN ('Acteur','Réalisateur','Producteur','Cameraman')) NOT NULL,

UNIQUE(Id\_film, Id\_personne, Role),

FOREIGN KEY(Id\_film) REFERENCES Film(Id\_film),

FOREIGN KEY(Id\_personne) REFERENCES Personne(Id\_personne)

);

CREATE TABLE Utilisateur(

Id\_user INTEGER PRIMARY KEY,

Id\_personne INTEGER UNIQUE,

Id\_localisation INTEGER,

Id\_compte INTEGER UNIQUE,

Email VARCHAR(50) NOT NULL,

Mdp VARCHAR(50) NOT NULL,

Telephone VARCHAR(50),

UNIQUE(Email,Mdp),

FOREIGN KEY(Id\_personne) REFERENCES Personne(Id\_personne),

FOREIGN KEY(Id\_compte) REFERENCES Compte(Id\_compte) ON DELETE CASCADE,

FOREIGN KEY(Id\_localisation) REFERENCES Localisation(Id\_localisation),

);

CREATE TABLE Compte(

Id\_compte INTEGER PRIMARY KEY,

Solde NUMERIC(10,2) CHECK( Solde > Decouvert\_autorise) NOT NULL,

Decouvert\_autorise NUMERIC(10,2) NOT NULL DEFAULT 0

);

CREATE TABLE Localisation(

Id\_localisation INTEGER PRIMARY KEY,

Latitude INTEGER,

Longitude INTEGER,

FOREIGN KEY(Latitude) REFERENCES Lieu(Latitude),

FOREIGN KEY(Longitude) REFERENCES Lieu(Longitude)

);

CREATE TABLE Lieu(

Latitude INTEGER,

Longitude INTEGER,

Departement VARCHAR(50) NOT NULL,

Ville VARCHAR(50) NOT NULL,

Pays VARCHAR(50) NOT NULL,

Adresse VARCHAR(50),

Code\_postale VARCHAR(6) NOT NULL,

PRIMARY KEY(Latitude, Longitude)

);

CREATE TABLE Festival(

Id\_festival INTEGER PRIMARY KEY,

Theme VARCHAR(50),

Nom\_festival VARCHAR(50) NOT NULL UNIQUE,

Capacite\_max INTEGER,

Description\_lieu VARCHAR(50),

Periodicite VARCHAR(50),

Date\_creation DATE,

Website VARCHAR(50) UNIQUE

);

CREATE TABLE Edition(

Id\_edition INTEGER PRIMARY KEY,

Id\_festival INTEGER,

Annee INTEGER,

Capacite\_max\_place INTEGER ,

Date\_debut DATE,

Date\_fin DATE,

Id\_localisation INTEGER,

UNIQUE(Id\_festival, Annee, Date\_debut, Date\_fin),

FOREIGN KEY(Id\_festival) REFERENCES Festival(Id\_festival),

FOREIGN KEY(Id\_localisation) REFERENCES Localisation(Id\_localisation)

);

CREATE TABLE Place(

Id\_place INTEGER PRIMARY KEY,

Id\_user INTEGER,

Nom\_place VARCHAR(50) NOT NULL,

Prenom\_place VARCHAR(50) NOT NULL,

Id\_categorie INTEGER,

Numero\_place INTEGER,

UNIQUE(Id\_festival, Nom\_place, Prenom\_place),

UNIQUE(Id\_festival, Numero\_place),

FOREIGN KEY(Id\_user) REFERENCES Utilisateur(Id\_user),

FOREIGN KEY(Id\_categorie) REFERENCES Categorie(Id\_categorie)

);

CREATE TABLE Categorie(

Id\_categorie INTEGER PRIMARY KEY,

Prix INTEGER,

Descriptif VARCHAR(50),

Id\_edition INTEGER,

Capacite\_max INTEGER ,

FOREIGN KEY(Id\_edition) REFERENCES Edition(Id\_edition)

);

CREATE TABLE Site\_Critique(

Id\_site INTEGER PRIMARY KEY,

Nom VARCHAR(50) NOT NULL,

Lien VARCHAR(50) NOT NULL UNIQUE

);

CREATE TABLE Critique(

Id\_critique INTEGER PRIMARY KEY,

Id\_film INTEGER,

Id\_site INTEGER,

Note\_globale INTEGER CHECK( Note\_globale BETWEEN 0 AND 10) NOT NULL,

Avis\_general VARCHAR(50),

UNIQUE(Id\_film, Id\_site),

FOREIGN KEY(Id\_film) REFERENCES Film(Id\_film),

FOREIGN KEY(Id\_site) REFERENCES Site\_Critique(Id\_site)

);