

Les instructions du DDL

Mots clés : CREATE TABLE, ALTER TABLE, Contrainte d'intégrités, CONSTRAINT, PRIMARY KEY, FOREIGN KEY, REFERENCES, UNIQUE, CHECK, NULL

Etablir le schéma entités – relations – attributs (ERA)

A partir de l'interview d'Alfred le gérant d'un magasin de location de films proposer le schéma ERA : c'est-à-dire la définition des tables, des colonnes, des types de données pour ces colonnes et des différentes contraintes d'intégrités

Interview d'Alfred le responsable de la boutique.

Nos clients sont de plus en plus exigeant, cependant ils oublient également fréquemment de nous rendre les DVD qu'ils ont loués, à la date indiquée lors de leur location. Bien qu'il s'agisse d'un club dans lequel nos clients sont de confiance, nous souhaitons mieux suivre nos locations et l'ensemble des éléments mis à disposition de nos membres.

Lorsque les clients empruntent un ou plusieurs films, ils s'engagent sur une durée de location. Même si certains dépassent cette date limite, la très grande majorité d'entre eux la respecte. Ainsi lorsqu'un film n'est pas disponible à la location, il nous est possible d'indiquer la date de retour prévue, ceci afin d'éviter que les gens reviennent inutilement au magasin.

Pour chaque client, il nous est nécessaire de connaître son nom, prénom, son adresse et un numéro de téléphone. Nous utilisons ces informations en cas de non retour d'un ou plusieurs DVD 2 jours après la date prévue pour contacter le client et l'inviter à rendre son exemplaire au plus vite. Ce délai est réduit à 24 h dans le cas d'une nouveauté.

Un client ne peut pas emprunter plus de 5 films à la fois avec au maximum 2 nouveautés. Toutefois, moyennant une majoration, le client peut exceptionnellement emprunter 3 nouveautés. Cette autorisation n'est accordée qu'une fois par mois au maximum.

Pour nous, une nouveauté est un film qui est disponible à la location depuis moins de 3 mois. Passé ce délai, le film perd son statut de nouveauté. Un même film peut être disponible en plusieurs exemplaires. C'est généralement le cas pour les nouveautés, puis au fur et à mesure que le nombre d'emprunts pour ce titre diminue, nous réduisons le nombre d'exemplaire disponibles à la location.

Chaque exemplaire, est parfaitement identifié par son numéro d'inventaire. Il dispose également d'une place spécifique en magasin. Cette place est repérée par son numéro de travée, et son numéro d'étagère. Il est possible de disposer plusieurs films différents sur une même étagère. En général nous essayons de regrouper ensemble tous les exemplaires d'un même film. Cet emplacement physique peut varier au cours du temps car les nouveautés sont mises en avant et ne sont pas classé selon la catégorie du film.

Les catégories permettent de classer de façon grossière les films, c'est par exemple: horreur, aventure, comédie, western, enfant, documentaire,

Lorsqu'un client vient pour louer des films, nous commençons toujours par nous assurer qu'il a bien rendu tous les films empruntés et qu'il est à jour dans ses règlements. Nous créons alors une fiche sur laquelle nous notons les références du client, la référence de chaque exemplaire emprunté, son titre et la date de retour prévue. Comme le montant de la location est lié à la durée d'emprunt, le client paiera lorsqu'il aura rendu tous les exemplaires empruntés. Le paiement s'effectue toujours en une seule fois.

Pour calculer le prix de location nous tenons compte des éléments suivant:

- la durée de location (en jours),
- le type du film : nouveauté ou non,
- le prix de location à la journée fixé au niveau de chaque film.

Nos clients et nous même souhaitons pour chaque film connaître le réalisateur et les principaux acteurs afin de pouvoir réaliser des recherches par rapport à ces critères.

Pour chaque film présent dans la boutique sous la forme de un ou plusieurs exemplaires nous dressons la fiche du film sur laquelle sont présent le détail du film, comme son nom, sa nationalité, son réalisateur, la liste des principaux acteurs, mais également un résumé.

Titre du Film



Les acteurs principaux :

.....

.....

Réalisé par :

Produit par :

Année de production :

Genre :

Nationalité :

Durée :

Catégorie de spectateurs :

Résumé :

.....

.....

.....

.....

.....

.....

Un e fois par mois, nous envoyons par mail, à nos clients qui possède une adresse électronique, la liste de nos nouveautés.

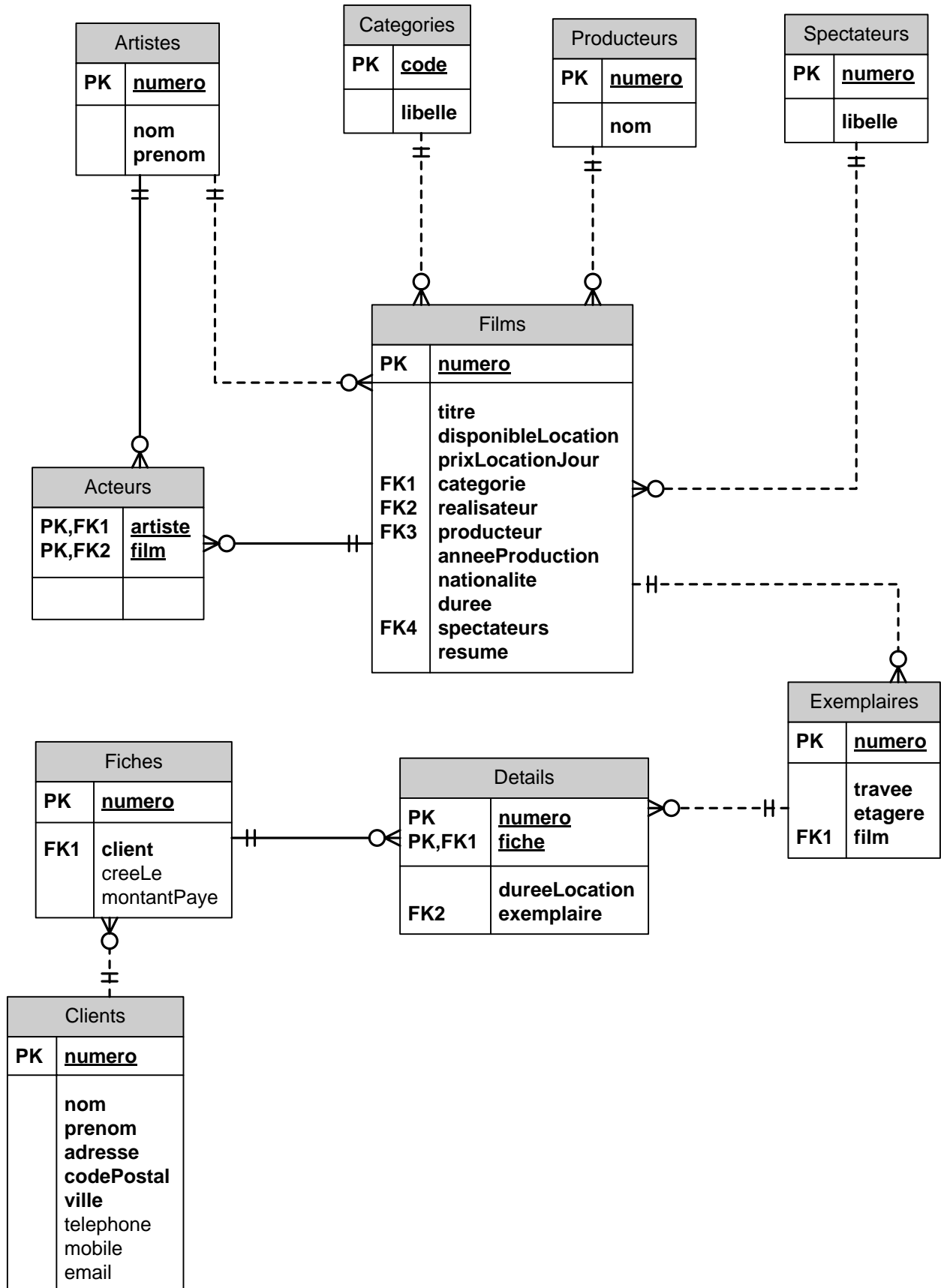
Identifier les entités

A la lecture de l'interview il est possible d'identifier les entités suivantes :

- Client : regroupe toutes les informations relatives au client
- Film : regroupe toutes les informations relatives à un film
- Artiste : regroupe les acteurs et les réalisateurs
- Exempleire : regroupe toutes les informations d'un exemplaire de film mis en location
- FicheLocation : regroupe toutes les informations communes à un ou plusieurs emprunts réalisés simultanément par un client
- DétailLocation : regroupe les informations spécifiques à chaque exemplaire emprunté
- Catégorie : permet de regrouper les films de façon logique

Le schéma ERA

Le schéma suivant donne un aperçu graphique du schéma ERA



Le détail des entités

Clients

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	identity
nom	nvarchar(50)	obligatoire	
prenom	nvarchar(50)	obligatoire	
adresse	nvarchar(100)	obligatoire	
codePostal	char(5)	obligatoire	
ville	nvarchar(50)	obligatoire	
telephone	char(14)		
mobile	char(14)		
email	nvarchar(100)		

Fiches

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	identity
client	int	obligatoire clé étrangère (Clients)	
creele	datetime		Date du jour par défaut
montantPaye	money		

Détails

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	
fiche	int	clé primaire clé étrangère (Fiches)	
dureeLocation	int	obligatoire	
exemplaire	int	obligatoire clé étrangère (Exemplaires)	

Exemplaires

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	identity
travee	int	obligatoire	
etagere	int	obligatoire	
film	int	obligatoire clé étrangère (Films)	

Films

Attribut	Type de données	Contraintes	Commentaire
numero	Int	clé primaire	identity
titre	nvarchar(80)	obligatoires	
disponibleLocation	Tinyint	obligatoire	
prixLocationJour	Money	obligatoire	
categorie	char(5)	obligatoire	
realisateur	int	obligatoire clé étrangère (Artistes)	
producteur	int	obligatoire clé étrangère (Producteurs)	
anneeProduction	smallint	obligatoire	
nationalite	char(5)	obligatoire	Valeur par défaut : Fra
duree	int	obligatoire	
spectateurs	int	obligatoire clé étrangère (Spectateurs)	
resume	nvarchar(max)	obligatoire	

Acteurs

Attribut	Type de données	Contraintes	Commentaire
artiste	int	clé primaire clé étrangère (Artistes)	
film	int	clé primaire clé étrangère (Films)	

Artistes

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	identity
nom	nvarchar(50)	obligatoire	
prenom	nvarchar(50)	obligatoire	

Categories

Attribut	Type de données	Contraintes	Commentaire
code	char(5)	clé primaire	
libelle	nvarchar(50)	obligatoire	

Producteurs

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	identity
nom	nvarchar(50)	obligatoire	

Spectateurs

Attribut	Type de données	Contraintes	Commentaire
numero	int	clé primaire	identity
libelle	nvarchar(50)	obligatoire	

Créer un nouveau projet

De nombreux scripts Transact SQL vont être définis, aussi est-il préférable de les regrouper dans un projet au sein de SQL Server Management Studio. Le projet portera le nom de **Video** et sera placé dans une nouvelle solution

Créer la base de données

Les entités identifiées, il va être possible de créer les tables dans SQL Server. Cependant, il n'est pas question de définir ces tables dans une base quelconque. Il est donc nécessaire, dans un premier temps, de définir la base de données **Video**. Pour définir cette base, il est préférable d'utiliser les caractéristiques suivantes :

- Fichier de données

Nom	Video_Data
Nom physique	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\Video_Data.mdf
Taille initiale	10 Mo
Croissance	10%
Taille maximale	50 Mo

- Fichier journal

Nom	Video_Log
Nom physique	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\Video_Log.ldf
Taille initiale	5 Mo
Croissance	10%
Taille maximale	Illimité

Le script suivant permet de créer une telle base :

```
use master;
go
CREATE DATABASE Video
ON PRIMARY (
    NAME = N'Video_Data',
    FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\Video_Data.mdf' ,
    SIZE = 10MB ,
    MAXSIZE = 50MB ,
    FILEGROWTH = 10%)
LOG ON (
    NAME = N'Video_log',
    FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\Video_log.ldf' ,
    SIZE = 5MB ,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 10%);
go
```

Définir un schéma

Afin de gagner en clarté certaines tables devront être définies dans le schéma Magasin tandis que les autres seront définies dans le schéma dbo.

Le schéma va être défini avec le script suivant :

```
use Video;
go
CREATE SCHEMA Magasin;
```

Créer les tables

Lors de cette étape de création de table, aucune contrainte d'intégrité n'est spécifiée à l'exception de la gestion des valeurs NULL (autorisées ou interdites). En procédant de cette façon, il n'y a pas d'ordre privilégié pour définir les tables. Les contraintes d'intégrités seront définies dans une seconde étape.

Les tables Clients, Fiches, Details et Exemplaires vont être définies dans le schéma Magasin. Les autres tables seront définies sur le schéma dbo.

```
use Video;
go
CREATE TABLE Spectateurs(
    numero int identity not null,
    libelle nvarchar(50) not null);
CREATE TABLE Producteurs(
    numero int identity not null,
    nom nvarchar(50) not null);
CREATE TABLE Categories(
    code char(5) not null,
    libelle nvarchar(50) not null);
CREATE TABLE Artistes(
    numero int identity not null,
    nom nvarchar(50) not null,
    prenom nvarchar(50) not null);
CREATE TABLE Acteurs(
    artiste int not null,
    film int not null);
CREATE TABLE Films(
    numero int identity not null,
    titre nvarchar(80) not null,
    disponibleLocation tinyint not null,
    prixLocationJour money not null,
    categorie char(5) not null,
    realisateur int not null,
    producteur int not null,
    anneeProduction smallint not null,
    nationalite char(5) not null default('Fra'),
    duree int not null,
    spectateur int,
    resume nvarchar(max)
);
CREATE TABLE Magasin.Exemplaires(
    numero int identity not null,
    travee int not null,
    etagere int not null,
    film int not null
```



```

);
CREATE TABLE Magasin.Details(
    numero int not null,
    fiche int not null,
    dureeLocation int not null,
    exemplaire int not null);
CREATE TABLE Magasin.Fiches(
    numero int identity not null,
    client int not null,
    creele datetime default getdate(),
    montantPaye money);
CREATE TABLE Magasin.Clients(
    numero int identity not null,
    nom nvarchar(50) not null,
    prenom nvarchar(50) not null,
    adresse nvarchar(100) not null,
    codePostal char(5) not null,
    ville nvarchar(50) not null,
    telephone char(14),
    mobile char(14),
    email nvarchar(100)
);

```

Définir les contraintes d'intégrités

Maintenant que les tables sont créées, il est possible de définir les contraintes d'intégrités. Pour réaliser cette opération dans de bonne condition, il est nécessaire de définir l'ensemble des contraintes de clé primaire (PK) dans un premier temps, puis les contraintes de clé étrangère(FK). Les contraintes d'unicité (UN) et de validation (CK) sont créées en dernier.

Contraintes de clé primaire

Chaque table possède une clé primaire. Cette clé peut être composée d'une ou plusieurs colonnes.

SQL Server utilise la structure d'index unique pour mettre en place cette contrainte. L'accès aux données à partir de la clé primaire est ainsi plus rapide. Par défaut c'est un index organisé (CLUSTERED) qui est défini. Les données sont donc physiquement organisées par rapport à la clé primaire.

Le tableau ci-dessous permet d'identifier pour chaque table les colonnes qui participent à la clé primaire ainsi que le nom de cette contrainte.

Table	Colonne	Nom de la contrainte
Clients	id	pk_clients
Fiches	numero	pk_fiches
Details	numero fiche	pk_details
Exemplaires	numero	pk_exemplaires
Films	numero	pk_films
Acteurs	artiste film	pk_acteurs
Artistes	id	pk_artistes
Categories	code	pk_categories
Producteurs	id	pk_producteurs

Genres	code	pk_genres
---------------	------	-----------

Les clés primaires sont mises en place avec le script suivant :

```
USE Video;
go
ALTER TABLE Magasin.Clients
  ADD CONSTRAINT pk_clients PRIMARY KEY(numero);
ALTER TABLE Magasin.Fiches
  ADD CONSTRAINT pk_fiches PRIMARY KEY (numero);
ALTER TABLE Magasin.Details
  ADD CONSTRAINT pk_details PRIMARY KEY (numero, fiche);
ALTER TABLE Magasin.Exemplaires
  ADD CONSTRAINT pk_exemplaires PRIMARY KEY (numero);
ALTER TABLE Films
  ADD CONSTRAINT pk_films PRIMARY KEY (numero);
ALTER TABLE Acteurs
  ADD CONSTRAINT pk_acteurs PRIMARY KEY (artiste, film);
ALTER TABLE Artistes
  ADD CONSTRAINT pk_artistes PRIMARY KEY (numero);
ALTER TABLE Categories
  ADD CONSTRAINT pk_categories PRIMARY KEY (code);
ALTER TABLE Producteurs
  ADD CONSTRAINT pk_producteurs PRIMARY KEY (numero);
ALTER TABLE Spectateurs
  ADD CONSTRAINT pk_spectateurs PRIMARY KEY(numero);
```

Contraintes de clés étrangères

Le tableau suivant présente les différentes contraintes de références qui existent dans le schéma étudié.

Colonnes de la clé étrangère	Colonnes référencées	Nom de contrainte
Films.categorie	Categories.code	fk_films_categories
Films.realisateur	Artistes.numero	fk_films_realisateur
Films.producteur	Producteurs.numero	fk_films_producteurs
Films.spectateur	Spectateurs.numero	fk_films_spectateurs
Acteurs.artiste	Artistes.numero	fk_acteurs_artistes
Acteurs.film	Films.numero	fk_acteurs_films
Magasin.Exemplaires.film	Films.numero	fk_exemplaires_films
Magasin.Details.exemplaire	Magasin.Exemplaires.numero	fk_details_exemplaires
Magasin.Details.fiche	Magasin.Fiches.numero	fk_details_fiches
Magasin.Fiches.client	Magasin.Clients.numero	fk_fiches_clients

Le script suivant permet de définir les différentes contraintes de références dont les caractéristiques sont exposées ci-dessus.

```
use Video;
go
ALTER TABLE Films
  ADD CONSTRAINT fk_films_categories
  FOREIGN KEY(categorie) REFERENCES Categories(code);
ALTER TABLE Films
  ADD CONSTRAINT fk_films_realisateur
  FOREIGN KEY(realisateur) REFERENCES Artistes(numero);
ALTER TABLE Films
  ADD CONSTRAINT fk_films_producteurs
  FOREIGN KEY(producteur) REFERENCES Producteurs(numero);
ALTER TABLE Films
```

```

ADD CONSTRAINT fk_films_spectateurs
FOREIGN KEY (spectateur) REFERENCES Spectateurs (numero);
ALTER TABLE Acteurs
ADD CONSTRAINT fk_acteurs_artistes
FOREIGN KEY (artiste) REFERENCES Artistes (numero);
ALTER TABLE Acteurs
ADD CONSTRAINT fk_acteurs_films
FOREIGN KEY (film) REFERENCES Films (numero);
ALTER TABLE Magasin.Exemplaires
ADD CONSTRAINT fk_exemplaires_films
FOREIGN KEY (film) REFERENCES Films (numero);
ALTER TABLE Magasin.Details
ADD CONSTRAINT fk_details_exemplaires
FOREIGN KEY (exemplaire) REFERENCES Magasin.Exemplaires (numero);
ALTER TABLE Magasin.Details
ADD CONSTRAINT fk_details_fiches
FOREIGN KEY (fiche) REFERENCES Magasin.Fiches (numero);
ALTER TABLE Magasin.Fiches
ADD CONSTRAINT fk_fiches_clients
FOREIGN KEY (client) REFERENCES Magasin.Clients (numero);

```

Contraintes de validation

Les contraintes de validations (CHECK) vont permettre de définir un certain nombre de règles simples de gestion des données. Il est important de définir ces règles au plus proche des données. En travaillant ainsi, il n'est pas possible d'outrepasser ces règles et donc les données sont toujours cohérentes. Le tableau ci-dessous présente les différentes contraintes de validation à définir.

Table	Description	Nom de la contrainte
Films	disponibleLocation={0,1}	ck_films_disponibleLocation
Films	prixLocationJour>=0	ck_films_prixLocationJour
Films	anneeProduction>=1900	ck_films_anneeProduction
Fiches	montantPaye contient null ou bien une valeur >=0	ck_fiches_montantPaye

Remarque : Certaines règles ne peuvent pas être définies sous forme de contrainte de validation. Un déclencheur de base de données sera donc défini ultérieurement.

Le script suivant permet d'ajouter ces différentes contraintes :

```

USE Video;
go
ALTER TABLE Films
ADD CONSTRAINT ck_films_disponibleLocation
CHECK (disponibleLocation IN (0,1));
ALTER TABLE Films
ADD CONSTRAINT ck_films_prixLocationJour
CHECK (prixLocationJour>=0);
ALTER TABLE Films
ADD CONSTRAINT ck_films_anneeProduction
CHECK (anneeProduction>=1900);
ALTER TABLE Magasin.Fiches
ADD CONSTRAINT ck_fiches_montantPaye
CHECK ((montantPaye is null) OR (montantPaye>=0));

```

Modifier la structure en ajoutant des colonnes

La structure actuelle pose des soucis en ce qui concerne le retour des exemplaires empruntés. En effet la durée de location correspond elle à une durée effective de location ou bien prévisionnelle ?

Et comment est-il possible d'identifier les gens qui rendent leur location systématiquement en retard ? Pour pouvoir répondre à l'ensemble de ces contraintes, il est nécessaire de modifier la structure actuelle de la table Details en supprimant la colonne dureeLocation et en ajoutant les colonnes retourPrevuLe et retourEffectifLe. Ces deux colonnes sont de type datetime.

Le script suivant permet de réaliser cette modification

```
USE Video;
go
-- Ajouter les colonnes
ALTER TABLE Magasin.Details
  ADD retourPrevuLe datetime;
ALTER TABLE Magasin.Details
  ADD retourEffectifLe datetime;
-- Supprimer une colonne
ALTER TABLE Magasin.Details
  DROP COLUMN dureeLocation;
```

Autre point à prendre en compte : la qualité des exemplaires mis à disposition. Bien que le supports numériques offrent une meilleure qualité d'image, le support manipulé par de nombreuses personnes n'est pas toujours irréprochable au niveau de la qualité. La qualité de chaque exemplaire doit donc être conservée. Une colonne qualité de type char(2) est donc ajoutée à la table des exemplaires. Les valeurs autorisées pour cette colonne sont : EX(excellent), TB(très bon), B(bon), M(moyen), F(faible), ME(médiocre). Le niveau de qualité observé est à relativiser par rapport à la date d'achat d'un exemplaire. Il est donc nécessaire d'ajouter une seconde colonne nommée dateAchat et de type datetime, afin de stocker la date d'acquisition de l'exemplaire.

Le script suivant permet de réaliser ces modifications

```
ALTER TABLE Magasin.Exemplaires
  ADD qualite char(2);
ALTER TABLE Magasin.Exemplaires
  ADD constraint ck_exemplaires_qualite
  CHECK (qualite IN ('EX', 'TB', 'B', 'M', 'F', 'ME'));
ALTER TABLE Magasin.Exemplaires
  ADD dateAchat datetime;
```

Poser des index

Les index vont permettre d'accéder plus rapidement aux différentes informations. Cependant la mise à jour des index représente un coût non négligeable pour le moteur de base de données. Il convient donc de définir des index sur les colonnes fréquemment impliquées dans des restrictions et des tris. Dans un premier temps il est préférable de travailler 'a minima'. Ainsi seuls les colonnes participant à aux contraintes de clés étrangères seront indexés.

La description des différents index à définir est donnée dans le tableau suivant :

Nom de l'index	Table	Colonne
ix_films_categories	Films	categorie
ix_films_realisateurs	Films	realisateur
ix_films_producteurs	Films	producteur
ix_films_spectateurs	Films	spectateur

ix_acteurs_artistes	Acteurs	artiste
ix_acteurs_films	Acteurs	film
ix_exemplaires_films	Magasin.Exemplaires	film
ix_details_exemplaires	Magasin.Details	exemplaire
.ix_details_fiches	Magasin.Details	fiche
ix_fiches_clients	Magasin.Fiches	client

Remarque : Les index sont toujours associés à la table qui possède la ou les colonnes indexées. Il est donc normal que les index ne soient pas définis sur un schéma.

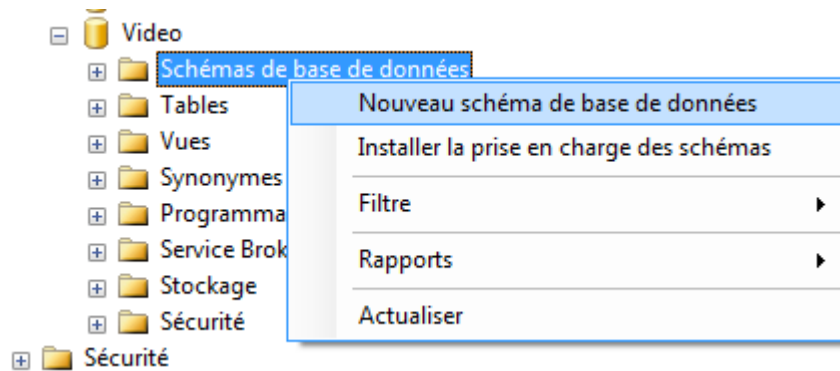
Le script suivant permet de réaliser la création des différents index :

```
use Video;
go
CREATE INDEX ix_films_categories
  ON Films(categorie);
CREATE INDEX ix_films_realisateurs
  ON Films(realisateur);
CREATE INDEX ix_films_producteurs
  ON Films(producteur);
CREATE INDEX ix_films_spectateurs
  ON Films (spectateur);
CREATE INDEX ix_acteurs_artistes
  ON Acteurs(artiste);
CREATE INDEX ix_acteurs_films
  ON Acteurs(film);
CREATE INDEX ix_exemplaires_films
  ON Magasin.Exemplaires(film);
CREATE INDEX ix_details_exemplaires
  ON Magasin.Details(exemplaire);
CREATE INDEX ix_details_fiches
  ON Magasin.Details(fiche);
CREATE INDEX ix_fiches_clients
  ON Magasin.Fiches(client);
```

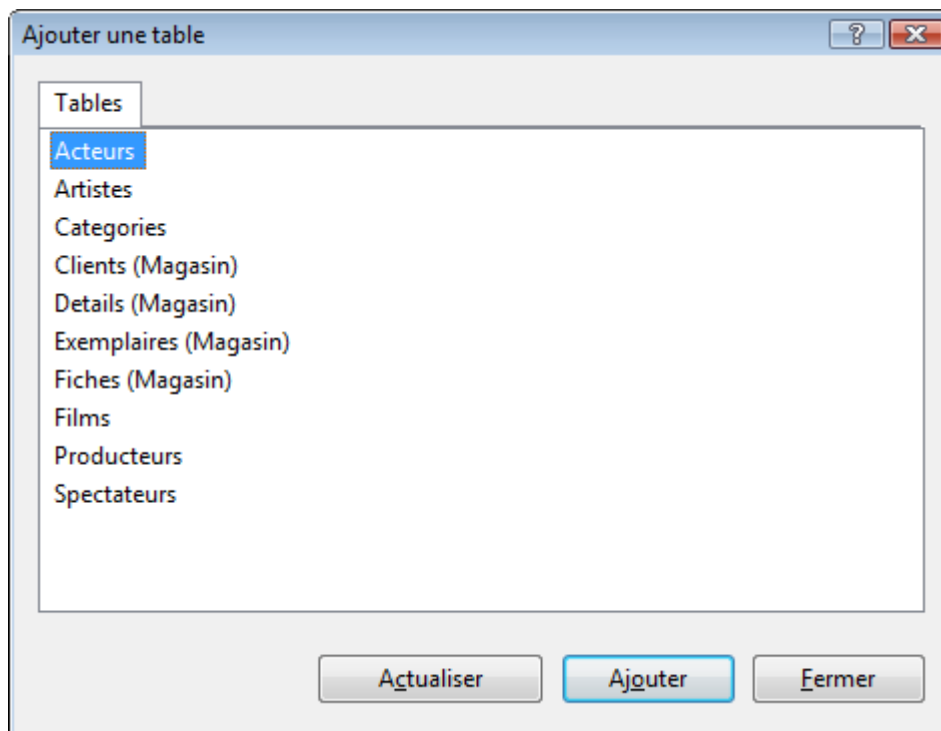
Etablir le diagramme de base de données

Maintenant que la structure de la base est stable, il est souhaitable d'avoir un aperçu visuel des tables, colonnes, types de données, contraintes de clés primaires et étrangères. Ce type de représentation permet d'avoir une vue d'ensemble de l'architecture des données. Lors de la conception de requêtes il est nécessaire d'avoir un tel schéma pour avoir quel sont les tables à inclure dans la requête et quels sont les jointures à effectuer.

Le schéma est défini depuis SQL Server Management Studio à partir du nœud **Schéma de base de données** dans l'explorateur de base de données.

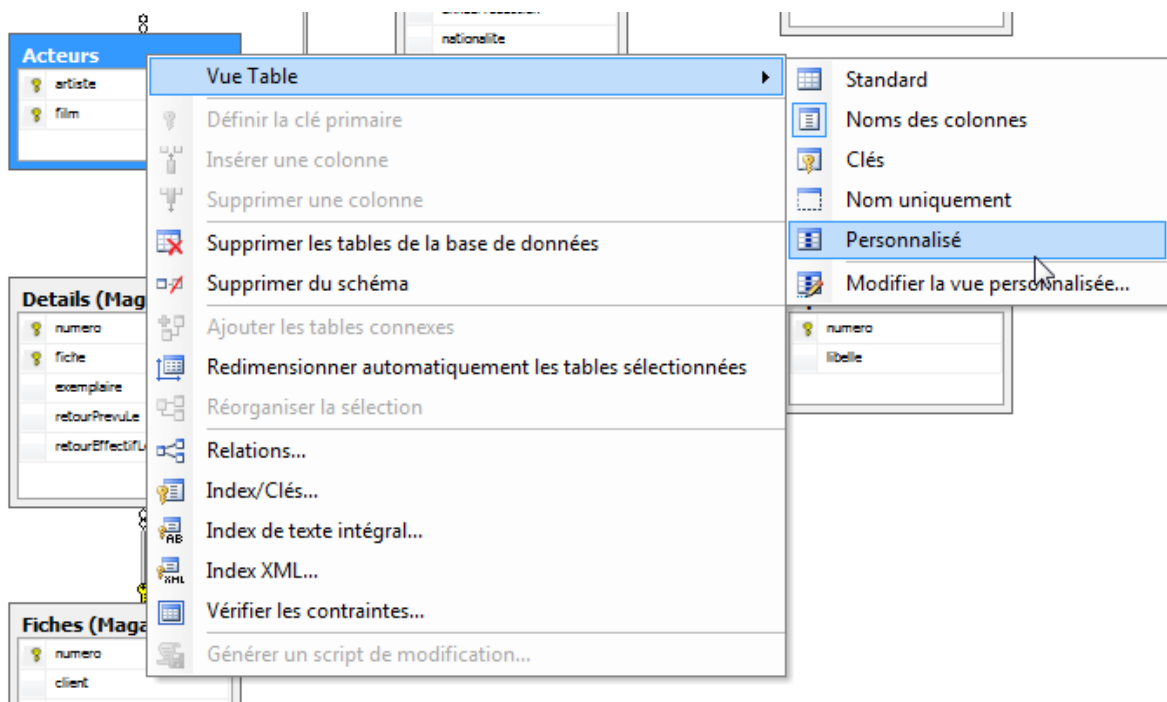


Après avoir accepté le message signalant l'activation des éléments nécessaire à l'établissement du schéma, il est possible de sélectionner les tables.



Après avoir sélectionnées toutes les tables elles sont ajoutées au schéma. Ce schéma est enregistré sous le nom Le schéma de la base de Vidéo.

Remarque : Si les informations présentes sur le schéma ne vous conviennent pas, il est possible de modifier la vue de chaque table à partir du menu contextuel associé à la table sur le schéma.



Définir des synonymes

Lors d'un échange avec Alfred, le responsable, il apparaît que la notion de clients dans la base le dérange et qu'il préfère celle de **membre** afin de développer l'aspect Club.

La mise en place de la structure est déjà bien avancée et il n'est pas question de tout modifier pour un simple problème de vocabulaire. De plus des développements ont peut être déjà commencés avec la table des Clients. Aussi pour donner satisfaction au client en adoptant son vocabulaire, il est possible de définir un synonyme afin de donner un second nom à la table des clients.

Le synonyme **Realisateurs** est également défini pour la table des Artistes. Ainsi les requêtes seront plus faciles à écrire et à comprendre car les rôles de réalisateurs et d'acteurs sont clairement identifiés.

La définition des synonymes est effectuée avec le script suivant :

```
USE Video;
go
CREATE SYNONYM Magasin.Membres FOR Magasin.Clients;
CREATE SYNONYM Realisateur FOR Artistes;
```