

Secteur Tertiaire Informatique
Filière « Développement Logiciel »

Séquence « Concevoir une base de données »

Solution MCD Cas Vidéo – SGBD MySQL

Apprentissage

Mise en pratique

Evaluation

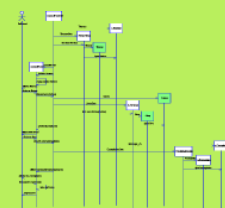
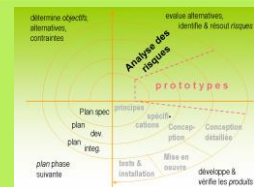
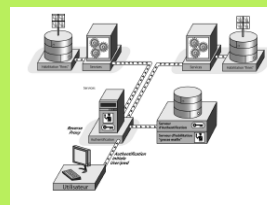


TABLE DES MATIERES

Table des matières	2
1. Dictionnaire de données.....	4
2. MCD de base	5
3. MCD Complet	6

Préambule

Ce document apporte une réponse au besoin de conception de la base de données du Cas Vidéo, au niveau du dictionnaire et du MCD.

Objectifs

Comme il n'existe pas de solution idéale dans ce domaine, on propose ici 2 variantes de MCD. Le plus important est alors de comparer votre solution avec celles qui sont proposées ici de manière à identifier les écarts et les erreurs éventuelles.

Méthodologie

1. DICTIONNAIRE DE DONNEES

La lecture attentive du cahier des charges exprimant les besoins de 'NICE VIDEO' permet d'établir une première liste de données à gérer dans la base de données (pour lever les ambiguïtés sur le sens, on s'appuie sur des exemples issus de l'énoncé) :

- Libellé du type de film ('dessin animé'...)
- Libellé du support ('DVD'...)
- Titre du film ('Seigneur des Anneaux'...)
- Code du film ('ANNEU1'...)
- Numéro d'exemplaire ('01'...)
- Numéro d'adhérent
- Justificatif de domicile adhérent
- Nom, Prénom, Adresse... de l'adhérent (à préciser)
- Etat de l'exemplaire (par exemple : 'P' pour perdu, 'R' pour rayé, 'O' pour OK...)
- Montant de la location (donnée calculée)
- Durée de la location (donnée calculée)
- Date de début de location de l'exemplaire
- Date du retour de l'exemplaire
- Prix par jour de location (dépend du type de support et du type de film)
- Nom et prénom du réalisateur, d'un acteur, d'un compositeur
- Rôle d'un participant à un film

On ajoutera l'année du film qui est une caractéristique implicite et on compétera les coordonnées client/adhérent (de plus, elles sont citées dans le dictionnaire provisoire fourni).

Concernant la participation d'une personne à un film, on peut considérer qu'une même personne peut avoir le 'rôle' d'acteur, de réalisateur ou de compositeur selon le film (Clint Eastwood peut même cumuler les 3 fonctions !) ; en conséquence, on identifiera des 'types de rôle' et on reliera les 'star' aux films selon leur(s) participation(s).

On ajoutera aussi les autres identifiants déjà listés dans le dictionnaire provisoire :

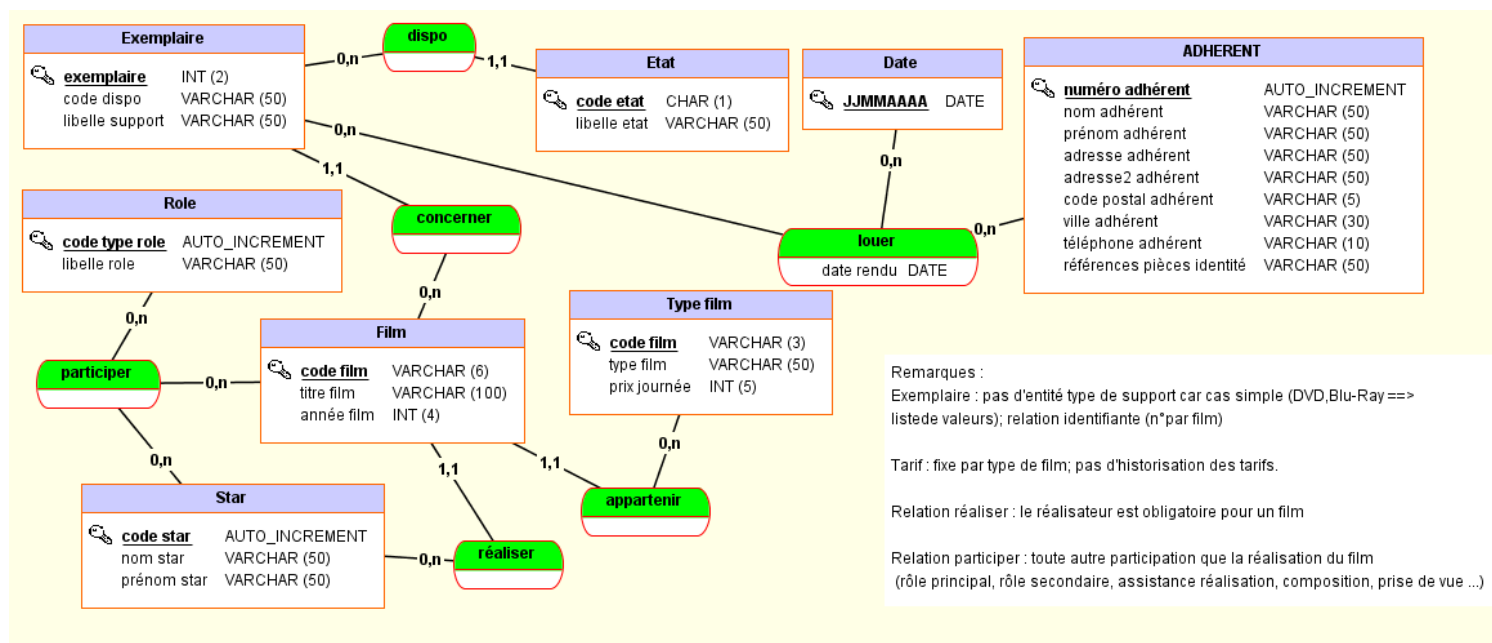
- Code de type de film
- Code de star

En complétant le tableau des contraintes fourni, on peut aboutir au dictionnaire suivant :

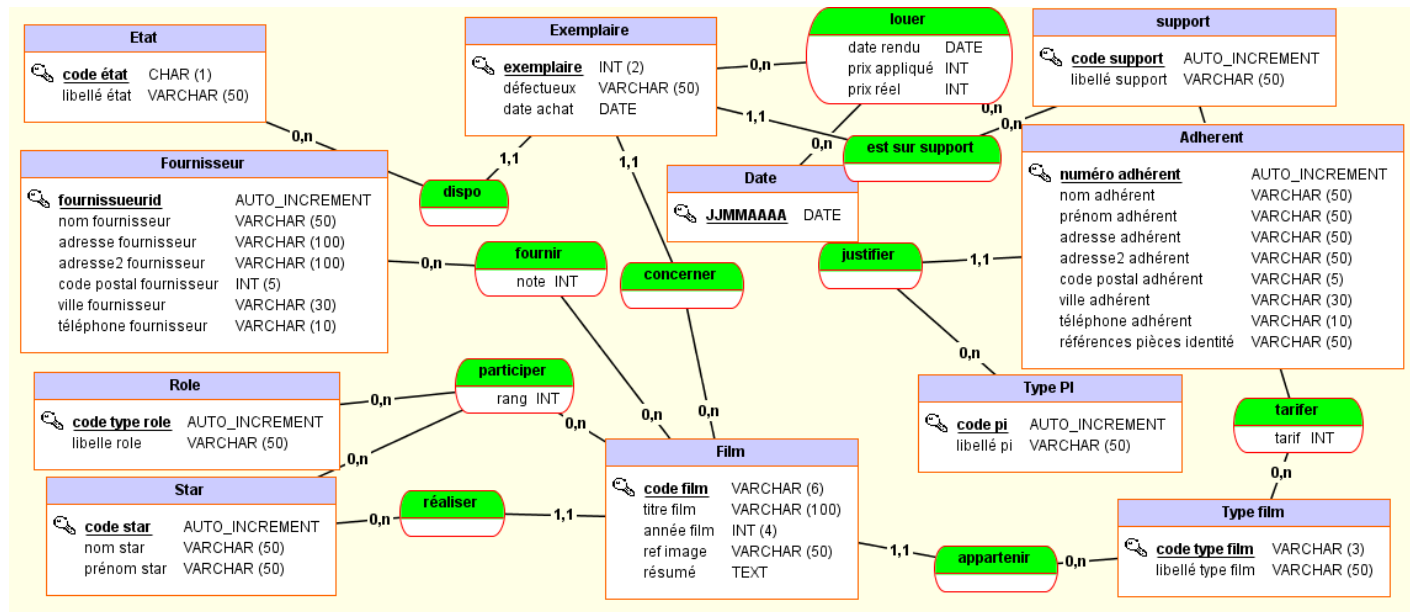
Attribut	Type de donnée	Longueur
Type de film	Car.	50
Libellé type de support	Car.	50
Code type de film	Car. Ou entier	3
Code film	Car.	6
Numéro ex.	Entier	2
Libellé/Titre film	Car.	100
Année film	Entier	4

Code Star	Entier	2
Nom Prénom Star	Car.	50
Code client/N° adhérent	Car. ou entier	4
Nom client	Car.	50
Prénom client	Car.	50
Adresse1 client	Car.	50
Adresse2 client	Car.	50
Code postal client	Car.	5
Ville client	Car.	30
Téléphone client	Car.	10
Réf pièce identité/Justificatif domicile	Car.	50
Date location	Date	
Date rendu	Date	
Prix journée	Entier	5
Etat de l'exemplaire	Car.	1
Code type de rôle	Entier	2
Libellé type de rôle	Car.	50

2. MCD DE BASE



3. MCD COMPLET



REMARQUES

Exemplaires : les différents états sont catalogués ainsi que les supports ; un booléen permet de savoir immédiatement si l'exemplaire est défectueux, donc impropre à la location (au prix d'une redondance)

Fournisseurs : extension non demandée explicitement dans le sujet

Tarification : prix de base variable par type de support (sans historisation) ; tarif appliqué pour chaque location et prix réel payé ==> larges possibilités de remise à partir de prix suggérés par le système

Justificatif : les types de pièces identité sont catalogués et on saisit les références du document présenté

1 -

```
CREATE INDEX ADHERENT_NOM_PRENOM ON ADHERENT (NOM_ADHERENT ASC, PRENOM_ADHERENT ASC) ;
```

2 - Au préalable il faut créer une entrée dans la table Type_pi :

```
INSERT INTO type_pi(libelle_pi) VALUES ('essai') ;
```

Si ça n'a pas été prévu à la création de la Table **adhèrent** il est toujours temps de spécifier une valeur par défaut du code postal (06000, oui c'est bien à Nice que ça se passe !!!) s'il n'est pas renseigné sur une requête d'insertion.

```
ALTER TABLE adherent MODIFY code_postal_adherent VARCHAR(5) NOT NULL DEFAULT '06000' ;
```

Ensuite, il suffit d'écrire le code du trigger qui vérifie avant l'insertion d'un enregistrement que le code postal est bien compris entre '01000' et '99999' !!!

```
CREATE TRIGGER `before_insert_adherent`
BEFORE INSERT ON `adherent` FOR EACH ROW
BEGIN
    IF (New.code_postal_adherent NOT BETWEEN '01000' AND '99999')
    THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Erreur : le
        Code postal n\'est pas valide';
    END IF;
END
```

Cette requête ne passe pas :

```
INSERT into adherent(code_pi,nom_adherent,prenom_adherent, code_postal
adherent,references_pieces_identite) VALUES (1,'Martin','Martin', '00
000','fgh5652552') ;
```

Alors que celle-ci est OK :

```
INSERT into adherent(code_pi,nom_adherent,prenom_adherent, code_postal
adherent,references_pieces_identite) VALUES (1,'Martin','Martin', '01
001','fgh5652552') ;
```

3 - Création d'un Trigger pour empêcher que la date de rendu soit inférieure à la date d'emprunt

```
CREATE TRIGGER `before_insert_louer`
BEFORE INSERT ON `louer` FOR EACH ROW
BEGIN
    IF (NEW.JJMMAAAA >= NEW.date_rendu)
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Erreur : la
    Date d\'emprunt doit être inférieure à la date de retour';
    END IF;
END
```

Pour pouvoir tester l'efficacité de cette contrainte il faut déjà créer :

```
INSERT into adherent(code_pi,nom_adherent,prenom_adherent, code_postal
adherent,references_pieces_identite) VALUES (1,'Dupont','Martin', '06
000','dupmart0001') ;
```

```
INSERT into etat values(1,'P') ;
```

```
INSERT into etat values(2,'R') ;
```

```
INSERT into etat values(3,'O') ;
```

Solution MCD Cas Vidéo

Afpa © 2016 – Section Tertiaire Informatique – Filière « Développement Logiciel »

```

INSERT into support(libelle_support) VALUES ("DVD") ;
INSERT into support(libelle_support) VALUES ("Blu-Ray") ;
insert into exemplaire (code_etat,code_film, code_support, date_achat,
exemplaire) VALUES (1, 'volau1', 1, '2018/02/10', 1) ;

INSERT into date VALUES ('2018/08/12') ;

```

Finalement, cette requête va fonctionner :

```

INSERT into louer(exemplaire,JJMMAAAA,numero_adherent,date_rendu) VALU
ES (1, '2018/08/12',5, '2018/08/14') ;

```

Alors que celle-ci ne passera pas :

```

INSERT into louer(exemplaire,JJMMAAAA,numero_adherent,date_rendu) VALU
ES (1, '2018/08/16',5, '2018/08/14') ;

```

4 - DROP TABLE IF EXISTS `etat`;

```

CREATE TABLE IF NOT EXISTS `etat` (
  `code_etat` ENUM('P','R','O') NOT NULL,
  `libelle_etat` varchar(50) NOT NULL,
  PRIMARY KEY (`code_etat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

5 - Pour répondre à la contrainte **annee_film** comprise entre 1900 et 2200 on peut utiliser l'instruction SQL **SIGNAL** qui permet de renvoyer des erreurs.

Ecriture du Trigger :

```

CREATE TRIGGER `before_insert_film`
BEFORE INSERT ON `film` FOR EACH ROW
BEGIN
  IF (NEW.annee_film IS NULL or NEW.annee_film NOT BETWEEN 1900 and
2200)
    THEN SIGNAL SQLSTATE'45000' SET MESSAGE_TEXT = 'Erreur :
1\'année du film doit être comprise entre 1900 et 2300';
  END IF;
END

```


Ainsi, dès qu'on voudra insérer une année de film qui n'appartient pas à 1900 et 2200 l'insertion sera impossible !!!

Pour pouvoir tester l'efficacité de cette contrainte il faut d'abord créer :

```
INSERT INTO type_film VALUES ('vol', 'Comédie dramatique') ;
```

```
INSERT INTO star VALUES (1, 'Nicholson', 'Jack') ;
```

Cette requête ne passe pas !!!

```
INSERT into film (annee_film, code_film, code_star, code_type_film, ref_image, resume, titre_film) VALUES (2300, 'volau1', 1, 'vol', 'dddddd', 'blablabla', 'Vol au dessus d\'un nid de coucous') ;
```

... Alors que celle-ci est OK :

```
INSERT into film (annee_film, code_film, code_star, code_type_film, ref_image, resume, titre_film) VALUES (1968, 'volau1', 1, 'vol', 'dddddd', 'blablabla', 'Vol au dessus d\'un nid de coucous') ;
```

CREDITS

ŒUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

B. Hézard - Formateur

A. Restoueix - Formateur

Ch. Perrachon - Ingénieure de formation

Date de mise à jour : 01/10/18

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

Solution MCD Cas Vidéo

Afpa © 2016 – Section Tertiaire Informatique – Filière « Développement Logiciel »