|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mnémonique | Signification | Type | Longueur | Contraintes |
| customer\_id | Identifiant du client | N | 11 | obligatoire |
| customer\_firstname | Prénom du client | A | 32 | obligatoire |
| customer\_lastname | Nom du client | A | 32 | obligatoire |
| customer\_address | Adresse du client | AN | 255 | obligatoire |
| customer\_deposit | Emprunt du client | N | 5 (3.2) | obligatoire |
| cassette\_number | Numéro de la cassette | N | 5 | obligatoire |
| cassette\_title | Titre de la cassette | AN | 24 | obligatoire |
| cassette\_state | Etat de la cassette | A | 15 | obligatoire |
| movie\_id | Identifiant du film | N | 11 | obligatoire |
| movie\_actor | Liste d’acteurs du film | A | 255 | obligatoire |
| movie\_realisator | Réalisateur du film | A | 50 | obligatoire |
| movie\_duration | Durée du film | T |  | obligatoire |
| movie\_gender | Genre du film | A | 15 | obligatoire |
| movie\_public | Film public | AN | 50 | obligatoire |
| borrow\_number | Numéro de l’emprunteur | N | 11 | obligatoire |
| borrow\_date | Date de l’emprunt | D |  | YYYY-MM-DD |
| borrow\_date\_return | Date de retour | D |  | YYYY-MM-DD |

Règle de gestion

Un client emprunte une ou plusieurs cassettes

Une cassette peut être empruntée pas un ou plusieurs clients

Un client peut faire zéro ou plusieurs emprunts

Un emprunt peur être fait par zéro ou plusieurs clients

Le film est contenu sur une ou plusieurs cassettes

Une cassette contient un film

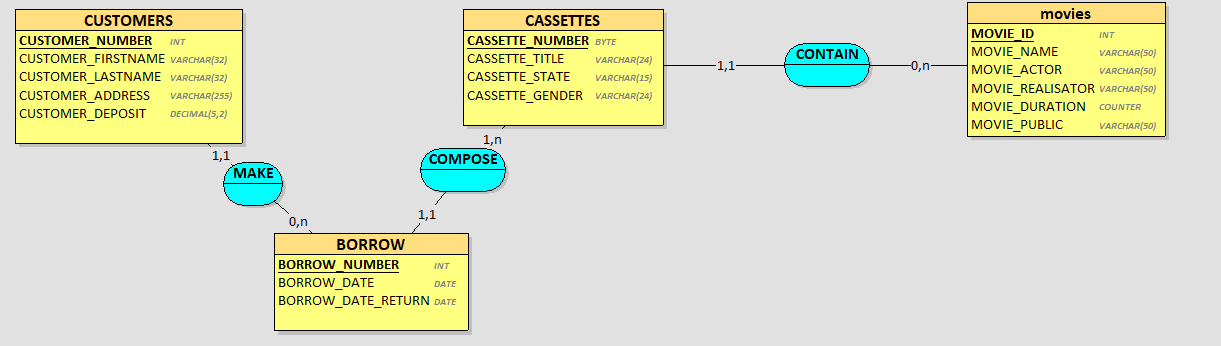
Dépendances fonctionnelles

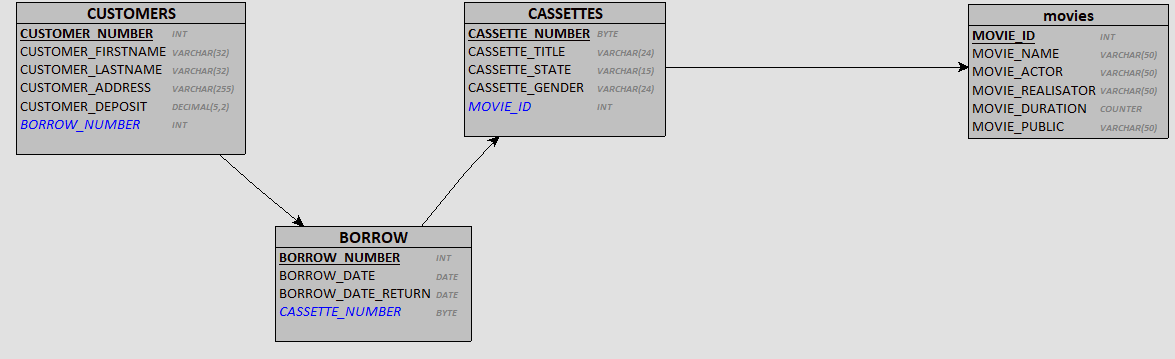
customer\_id **🡪** customer\_firstname, customer\_lastname, customer\_address, customer\_deposit

cassette\_number **🡪** cassette\_title, cassette\_state, cassette\_gender

borrow\_number **🡪** borrow\_date, borrow\_date\_return

movie\_id **🡪** movie\_actor, movie\_realisator, movie\_duration, movie\_gender, movie\_public

MCD LOOPING

MLD LOOPING

SQL Server

Création de table

-- j'utilise la base de donnée selectionnée

USE db\_video\_club;

-- je détruis la table si elle existe déja

DROP TABLE IF EXISTS customers;

DROP TABLE IF EXISTS borrow;

DROP TABLE IF EXISTS cassettes;

DROP TABLE IF EXISTS movies;

CREATE TABLE movies(

movie\_id INT NOT NULL,

movie\_name VARCHAR (50) NOT NULL,

movie\_actor VARCHAR (50) NOT NULL,

movie\_realisator VARCHAR (50) NOT NULL,

movie\_duration TIME NOT NULL,

movie\_public VARCHAR (50) NOT NULL,

-- on renomme la clé primaire movie\_id en PK\_Movies\_movie\_id voir cours page 4

CONSTRAINT PK\_Movies\_movie\_id PRIMARY KEY (movie\_id)

);

CREATE TABLE cassettes(

cassette\_number INT NOT NULL,

cassette\_title VARCHAR (24) NOT NULL,

cassette\_state VARCHAR (15) NOT NULL,

cassette\_gender VARCHAR (24) NOT NULL,

movie\_id INT,

-- on renomme la clé primaire voir cours page 4

CONSTRAINT PK\_Cassettes\_cassette\_number PRIMARY KEY (cassette\_number),

--Contrainte sur les clés etrangeres on renomme la clé d'une autre table voir cours page 5

CONSTRAINT FK\_Movies\_movie\_id FOREIGN KEY (movie\_id) REFERENCES movies (movie\_id)

);

CREATE TABLE borrow(

borrow\_number INT NOT NULL,

borrow\_date DATE NOT NULL,

borrow\_date\_return DATE NOT NULL,

cassette\_number INT,

-- on renomme la clé primaire voir cours page 4

CONSTRAINT PK\_Borrow\_borrow\_number PRIMARY KEY (borrow\_number),

--Contrainte sur les clés etrangeres on renomme la clé d'une autre table voir cours page 5

CONSTRAINT FK\_Cassettes\_cassette\_number FOREIGN KEY (cassette\_number) REFERENCES cassettes (cassette\_number),

);

CREATE TABLE customers(

customer\_number INT NOT NULL,

customer\_firstname VARCHAR (32) NOT NULL,

customer\_lastname VARCHAR (32) NOT NULL,

customer\_address VARCHAR (255) NOT NULL,

customer\_deposit DECIMAL (5,2) NOT NULL,

borrow\_number INT,

-- on renomme la clé primaire voir cours page 4

CONSTRAINT PK\_Customers\_customer\_number PRIMARY KEY (customer\_number),

--Contrainte sur les clés etrangeres on renomme la clé d'une autre table voir cours page 5

CONSTRAINT FK\_Borrow\_borrow\_number FOREIGN KEY (borrow\_number) REFERENCES borrow (borrow\_number),

) ;

Insertion de données

USE db\_video\_club;

INSERT INTO customers(customer\_number, customer\_firstname, customer\_lastname, customer\_address, customer\_deposit, borrow\_number)

VALUES (1, 'Sabah', 'Dekkoumi', '123 rue du soleil', 456.00, 123),

(2, 'Xavier', 'Bardon', '81 rue de la lune', 99.00, NULL)

INSERT INTO borrow(borrow\_number, borrow\_date, borrow\_date\_return, cassette\_number)

VALUES (123, '2022-02-14', '2022-02-21', 444),

(124, '2022-02-15', '2022-02-22', NULL);

-- marche bien

INSERT INTO cassettes (cassette\_number, cassette\_title, cassette\_state, cassette\_gender, movie\_id)

VALUES(444, 'dynastie', 'bon','comedie dramatique', 12),

(445, 'Demolition man', 'bon','Action', 13);

INSERT INTO movies (movie\_id, movie\_name, movie\_actor, movie\_realisator, movie\_duration, movie\_public)

VALUES (12,'equalizer', 'wesley', 'denis spielberg','01:30:00', 'tout public'),

(13,'equalizer 2', 'wesley', 'denis brogniart','01:45:00', 'tout public');

Sélection de données

--USE db\_video\_club;

CREATE VIEW customer\_cassette\_movies

AS

-- Selectionnée une entité (C grace a AS) et ces attributs, voir cours page 8

SELECT M.movie\_name, M.movie\_actor, M.movie\_realisator, M.movie\_duration, M.movie\_public,

CA.cassette\_title, CA.cassette\_state, CA.cassette\_gender,

B.borrow\_date, B.borrow\_date\_return,

C.customer\_number, C.customer\_firstname, C.customer\_lastname, C.customer\_deposit

-- FROM : dire depuis quelle entité AS : dans ce cas transforme l'entité customers en C, voir cours page 8

-- jointure entre cassette et movie

FROM movies AS M INNER JOIN cassettes AS CA

ON CA.movie\_id = M.movie\_id

-- Jointure entre cassette et borrow

INNER JOIN borrow as B

ON B.cassette\_number = CA.cassette\_number

-- Jointure customer et borrow

INNER JOIN customers AS C

ON C.borrow\_number = B.borrow\_number;

SELECT \* FROM customer\_cassette\_movies;

SELECT \* FROM movies;

SELECT \* FROM cassettes;

SELECT \* FROM borrow;

SELECT \* FROM customers;

SELECT customers.customer\_firstname, borrow.borrow\_date, movies.movie\_name

FROM customers INNER JOIN borrow ON customers.borrow\_number = borrow.borrow\_number

INNER JOIN cassettes ON borrow.cassette\_number = cassettes.cassette\_number

INNER JOIN movies ON cassettes.movie\_id = movies.movie\_id;

DROP VIEW customer\_cassette\_movies;