

| 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 par **un** ou **plusieurs** clients

Un client peut faire **zéro** ou **plusieurs** emprunts

Un emprunt peut être fait par **zéro** ou **plusieurs** clients

Le film est contenu sur **une** ou **plusieurs** cassettes

Une cassette contient **un** film

## Dépendances fonctionnelles

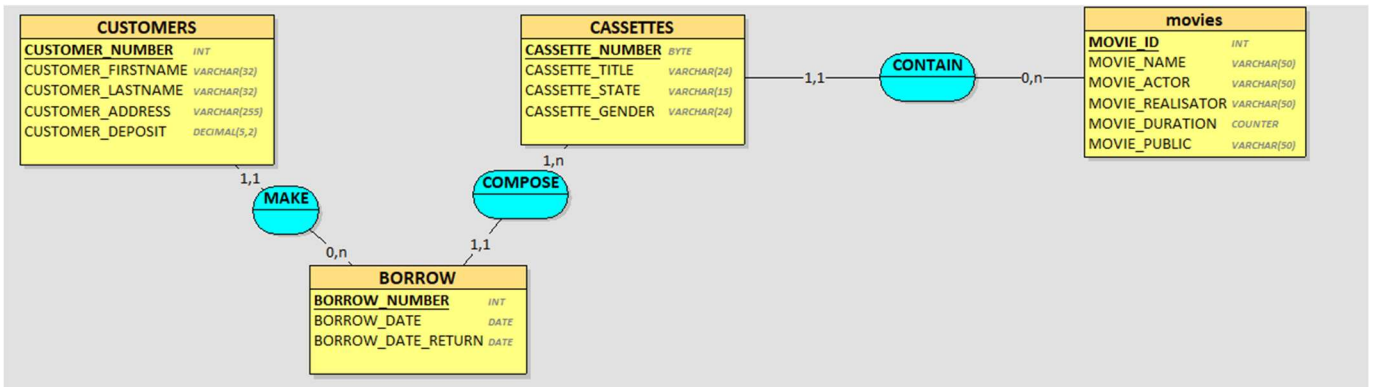
customer\_id → customer\_firstname, customer\_lastname, customer\_address, customer\_deposit

cassette\_number → cassette\_title, cassette\_state, cassette\_gender

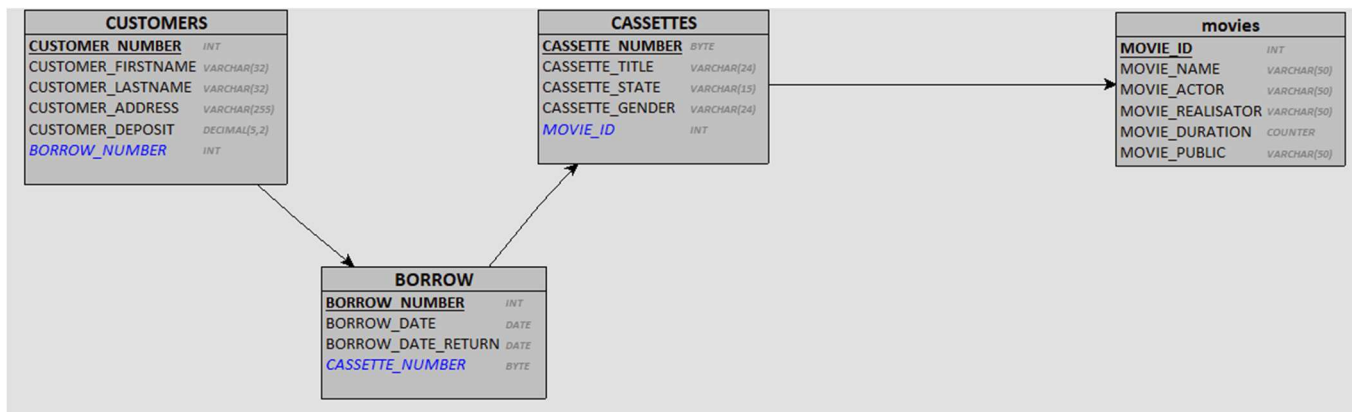
borrow\_number → borrow\_date, borrow\_date\_return

movie\_id → movie\_actor, movie\_realisator, movie\_duration, movie\_gender, movie\_public

## MCD LOOPING



## MLD LOOPING



---

*Création de table*

---

```
-- j'utilise la base de donnée selectionnée
USE db_video_club;
-- je détruis la table si elle existe déjà

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
-- Sélectionnée 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;
```