

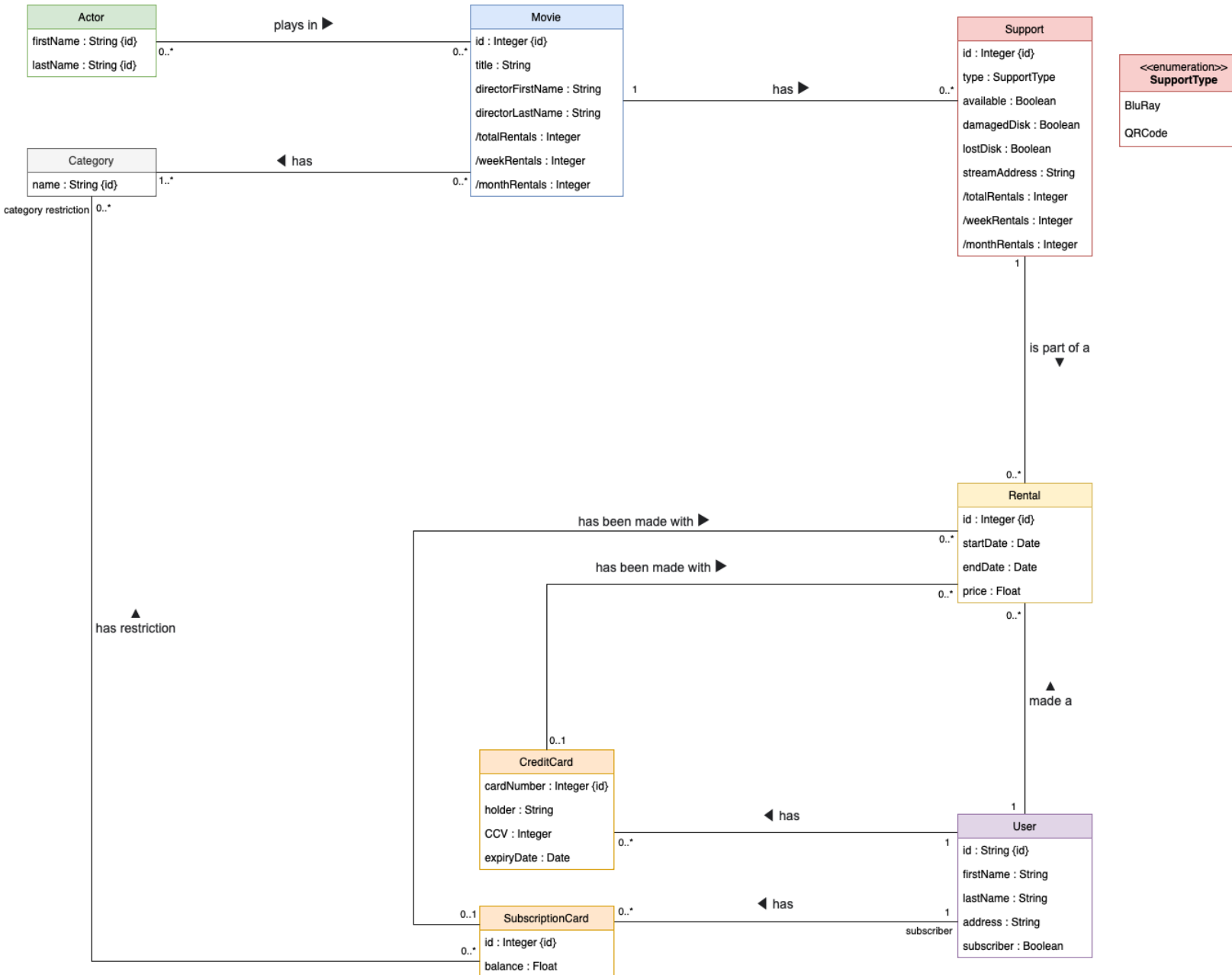
Schéma Conceptuel Base de Données

Groupe 2-C

BULTINCK Léonard MARCHIONINI David
GUTFREUND Delphine DA CRUZ SANTOS Bruno

December 12, 2022

1 Modèle Conceptuel



2 Modèle Relationnel

2.1 Schémas relationnels

Actors(firstName, lastName)

{<f, l> ∈ Actors \iff Un acteur est identifié par son prénom f et son nom l. }

Movies_base(movieID, title, directorFirstName, directorLastName)

{<i, t, f, l> ∈ Movies_base \iff Un film est identifié par un identifiant i. Son titre est t et son réalisateur a comme prénom f et comme nom l. }

(View) Movies(movieID, title, directorFirstName, directorLastName, totalRentals, weekRentals, monthRentals)

{<i, t, f, l, tr, wr, mr> ∈ Movies \iff Un film est identifié par un identifiant i. Son titre est t, son réalisateur a comme prénom f et comme nom l. Il a été loué tr fois au total, wr cette semaine et mr fois ce mois. }

ActorsMovies(firstName, lastName, movieID)

{<f, l, m> ∈ ActorsMovies \iff L'acteur dont le nom est l et le prénom est f joue dans le film identifié par m. }

Categories(categoryName)

{<n> ∈ Categories \iff Une catégorie est identifiée par son nom n. }

MoviesCategories(movieID, category)

{<m, c> ∈ MoviesCategories \iff Le film identifié par m est de catégorie c. }

Supports_base(supportID, type, available, damagedDisk, lostDisk, streamAddress, movieID)

{<i, t, a, d, l, s, m> ∈ Supports_base \iff Le support identifié par l'identifiant i est de type t, de disponibilité a, est endommagé si d est vrai, est perdu si l est vrai, et son adresse de streaming est s. Il est associé au film m. }

(View) Supports(supportID, type, available, damagedDisk, lostDisk, streamAddress, movieID, totalRentals, weekRentals, monthRentals)

{<i, t, a, d, l, s, m, tr, wr, mr> ∈ Supports \iff Le support identifié par l'identifiant i est de type t, de disponibilité a, est endommagé si d est vrai, est perdu si l est vrai, et son adresse de streaming est s. Il est associé au film m. Il a été loué tr fois au total, wr cette semaine et mr fois ce mois. }

Rentals(rentalID, startDate, endDate, price, userID, supportID, cardNumber, cardID)
 $\{ \langle i, d, e, p, u, s, cc, sc \rangle \in \text{Rentals} \iff \text{Une location est identifi  e par l'identifiant } i. \text{ L'utilisateur } u \text{ a lou   le support } s \text{    la date } d. \text{ La location se termine    la date } e \text{ et le prix de cette derni  re est } p. \text{ La location a   t   effectu  e avec la carte de cr  dit } cc \text{ ou la carte d'abonnement } sc. \}$

Users(userID, firstName, lastName, address, subscriber)
 $\{ \langle i, f, l, a, s \rangle \in \text{Users} \iff \text{L'utilisateur identifi   par l'identifiant } i, \text{ ayant pour pr  nom } f \text{ et pour nom } l \text{ et habitant    l'adresse } a \text{ est abonn   si } s \text{ est vrai.} \}$

CreditCards(cardNumber, holder, CCV, expiryDate, userID)
 $\{ \langle n, h, c, e, u \rangle \in \text{CreditCards} \iff \text{La carte de cr  dit identifi  e par le num  ro } n, \text{ appartient    } h, \text{ a comme CCV } c \text{ et expire    la date } expiryDate. \text{ Elle est utilis  e par l'utilisateur } u. \}$

SubscriptionCards(cardID, balance, userID)
 $\{ \langle i, b, u \rangle \in \text{SubscriptionCards} \iff \text{La carte identifi  e par l'identifiant } i \text{ a comme solde disponible } b. \text{ Elle est associ  e    l'utilisateur } u. \}$

CategoriesRestrictions(cardID, categoryName)
 $\{ \langle i, c \rangle \in \text{CategoriesRestrictions} \iff \text{La carte } c \text{ ne peut pas voir les films de cat  gorie } k. \}$

2.2 Description des domaines

domaine (firstName) = domaine (lastName) = domaine (title) = domaine (categoryName) =
 domaine (directorFirstName) = domaine (directorLastName) = domaine (streamAddress) =
 domaine (address) = domaine (userID) = cha  ne de caract  res

domaine (movieID) = domaine (totalRentals) = domaine (weekRentals) = domaine (monthRentals) =
 domaine (supportID) = domaine (rentalID) = domaine (cardNumber) = domaine (cardID) =
 domaine (CCV) = entiers ≥ 0

domaine (startDate) = domaine (endDate) = domaine (expiryDate) = Date

domaine (available) = domaine (damagedDisk) = domaine (lostDisk) = domaine (subscriber) =
 bool  ens

domaine (price) = r  els ≥ 0

domaine (balance) = r  els

domaine (type) = {“BluRay”, “QRCode” }

Contraintes d'intégrité référentielle

$\text{ActorsMovies}[\text{firstName}, \text{lastName}] \subseteq \text{Actors}[\text{firstName}, \text{lastName}]$

$\text{ActorsMovies}[\text{movieID}] \subseteq \text{Movies_base}[\text{movieID}]$

$\text{MoviesCategories}[\text{movieID}] = \text{Movies_base}[\text{movieID}]$

$\text{MoviesCategories}[\text{categoryName}] \subseteq \text{Categories}[\text{categoryName}]$

$\text{Supports}[\text{movieID}] \subseteq \text{Movies_base}[\text{movieID}]$

$\text{Rentals}[\text{supportID}] \subseteq \text{Supports}[\text{supportID}]$

$\text{Rentals}[\text{userID}] \subseteq \text{Users}[\text{userID}]$

$\text{Rentals}[\text{cardNumber}] \subseteq \text{CreditCards}[\text{cardNumber}]$

$\text{Rentals}[\text{cardID}] \subseteq \text{SubscriptionCards}[\text{cardID}]$

$\text{CreditCards}[\text{userID}] \subseteq \text{Users}[\text{userID}]$

$\text{SubscriptionCards}[\text{userID}] \subseteq \text{Users}[\text{userID}]$

$\text{CategoriesRestrictions}[\text{categoryName}] \subseteq \text{Categories}[\text{categoryName}]$

$\text{CategoriesRestrictions}[\text{cardID}] \subseteq \text{SubscriptionCards}[\text{cardID}]$