

# All the bad Movies

## Datenbanksysteme

### Meilenstein1a: Anforderungsanalyse für Kino-Portal

Jedes Kino hat einen eindeutigen Namen, ist zu unterschiedlichen Zeiten geöffnet und befindet sich an einer zusammengesetzten Adresse. Ein Kino besitzt in der Regel mehrere Säle. In jedem Kino kann zu einem Zeitpunkt nur eine Vorstellung eines Films in einem Saal spielen. Filme sind durch eine ISAN-Nummer eindeutig gekennzeichnet, haben aber auch einen Titel, ein Genre, ein Rating und werden von mehreren Darstellern besetzt. Ein Saal hat mehrere Sitzplätze in Reihe und Nummer aufgeteilt, die nur eindeutig innerhalb Saal definiert werden. Säle haben eine eindeutige Saalnummer und unterschiedliche Ausstattung wie, 48fps, 3D und IMAX und eine maximale Sitzplatzanzahl. Im Kino arbeiten mehrere Mitarbeiter, die über eine Personalnummer identifiziert werden, einen Namen, Gehalt, Positionsbezeichnung und eine hinterlegte Kontonummer haben. User können sowohl registriert, als auch nicht registriert Karten im System reservieren. Dabei wird eine temporäre Session-ID für die Darstellung und die IP-Adresse gespeichert. Registrierte Nutzer haben zusätzlich noch einen Benutzernamen und eine hinterlegte Mail-Adresse für die Anmeldung. User können Tickets für einen oder mehrere Sitzplätze reservieren. Dadurch werden Saalnummer, Kino, Film, Preis und Zeitpunkt definiert. Eine Reservierungsnummer wird dabei generiert. Eine Reservierung ist kein Kauf und verfällt, insofern die Karte nicht 30 Minuten vor Filmstart abgeholt wird.

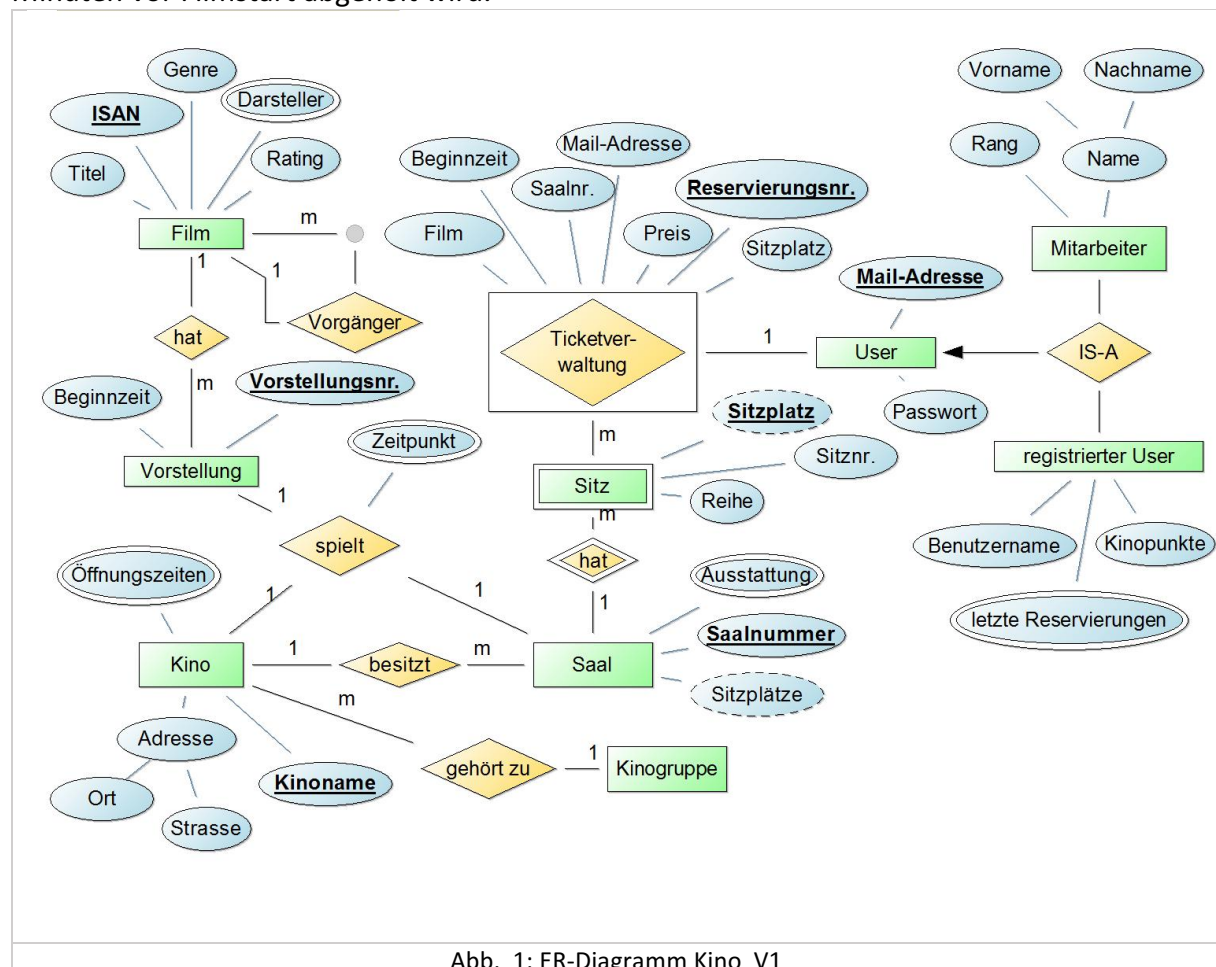


Abb. 1: ER-Diagramm Kino V1

## Meilenstein 1b: Anforderungsanalyse für Kino-Portal V2

Jedes Kino hat einen eindeutigen Namen, ist zu gewissen Zeiten geöffnet und befindet sich an einer Adresse. Ein Kino besitzt in der Regel mehrere Säle. In jedem Kino kann zu einem Zeitpunkt nur eine Vorstellung in einem Saal spielen. Filme sind durch eine ISAN-Nummer eindeutig gekennzeichnet, haben aber auch einen Titel, ein Rating und werden von mehreren Darstellern besetzt. Ein Saal hat mehrere Sitzplätze in Reihe und Nummer aufgeteilt, die nur eindeutig innerhalb des Saals definiert sind. Säle haben keine eindeutige Saalnummer, weil dem Kino zugeordnet, ist als barrierefrei gekennzeichnet, hat eine Leinwandfläche und eine maximale Sitzplatzanzahl. User können sowohl registrierte User, als auch angestellte sein. User müssen spätestens bei der Reservierung Name und Mailadresse angeben. Registrierte Nutzer haben zusätzlich noch einen Benutzernamen, Passwort und können Punkte für Filme sammeln. Mitarbeiter haben eine Mitarbeiternummer, einen Rang und restliche Urlaubstage. User können Tickets für einen oder mehrere Sitzplätze zu einer Vorstellung reservieren. Dadurch werden Saalnummer, Kino, Film, Sitzplatz und Zeitpunkt definiert. Eine Reservierungsnummer wird dabei generiert.

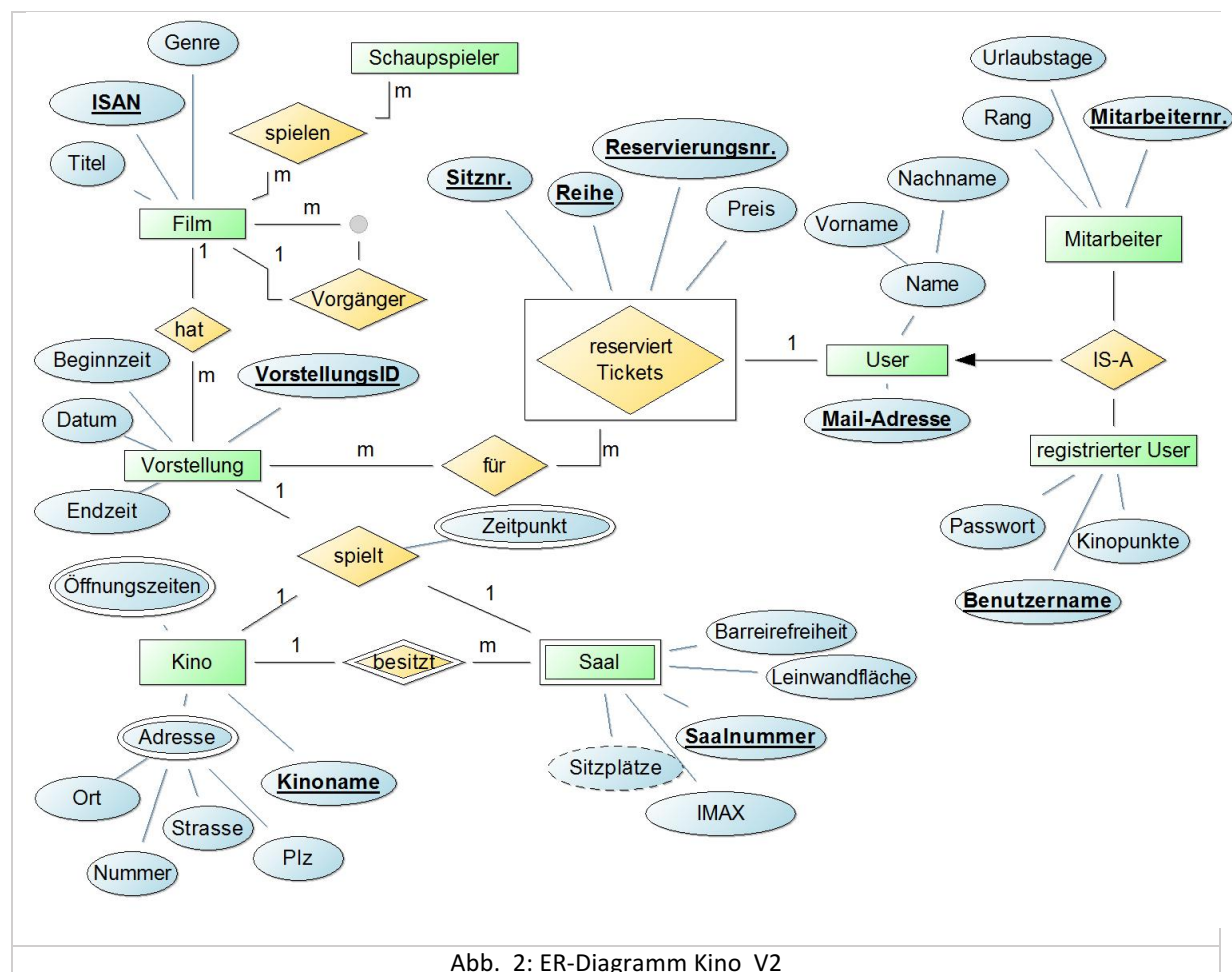


Abb. 2: ER-Diagramm Kino V2

## Meilenstein 2: Logischer Entwurf Relationenschema

tbl\_movie(ISAN, title, rating)  
PK: tbl\_movie.ISAN

tbl\_actor(ISAN, actor)  
PK: tbl\_actor.ISAN, actor  
FK: tbl\_actor.ISAN  $\diamond$  tbl\_movie.ISAN

tbl\_cinema(cinemaName, zipcode, city, street, number)  
PK: tbl\_cinema.cinemaName

tbl\_cinemaopeninghours(cinemaName, opentimes)  
PK: (cinemaName, opentimes)  
FK: tbl\_cinemaopeninghours.cinemaName  $\diamond$   
tbl\_cinema.cinemaName

tbl\_prequels(ISAN1, ISAN2)  
PK: (ISAN1, ISAN2)  
FK: tbl\_prequels.ISAN1  $\diamond$  tbl\_movie.ISAN  
FK: tbl\_prequels.ISAN2  $\diamond$  tbl\_movie.ISAN  
//CHECK ISAN1!= ISAN2

tbl\_show(showId, ISAN, showstart, length, agerestriction)  
--Date stored is showstart  
PK: (showId)  
FK: tbls\_show.ISAN  $\diamond$  tbls\_movie.ISAN

tbl\_hall(hallId, cinemaName, disaccess, screenarea)  
PK: (hallId, cinemaName)  
FK: tbl\_hall.cinemaName  $\diamond$  tbl\_cinema.cinemaName

tbl\_ticket(rowNr, seatNr, showId)  
PK: (rowNr, seatNr, showId)  
FK: tbl\_ticket.showId  $\diamond$  tbl\_show.showId

rel\_plays(showId, hallId, cinemaName)  
PK: (showId, hallId, cinemaName)  
FK: tbl\_plays.showId  $\diamond$  tbl\_show.showId  
FK: tbl\_plays.hallId  $\diamond$  tbl\_hall.hallId  
FK: tbl\_plays.cinemaName  $\diamond$  tbl\_cinema.cinemaName

isa\_user(mailaddress, firstname, lastname)  
PK: mailaddress

rel\_staff(staffId, mailaddress, rank, vacationdays)

PK: mailaddress

FK: rel\_staff.mailaddress ◇ isa\_user.mailaddress

rel\_reguser(username, mailaddress, password, moviepoints)

PK: mailaddress

FK: rel\_reguser.mailaddress ◇ isa\_user.mailaddress

rel\_reservation (reservationId, *mailaddress*, showId, hallId, cinemaName,  
rowNr, seatNr)

PK: (showId, hallId, cinemaName, rowNr, seatNr)

FK: rel\_reservation.mailaddress ◇ tbl\_user.mailaddress

rel\_reservation.showId ◇ tbl\_show.showId

rel\_reservation.hallId ◇ tbl\_show.hallId

rel\_reservation.cinema ◇ tbl\_show.cinemaName

rel\_reservation.rowNr ◇ tbl\_ticket.rowNr

rel\_reservation.seatNr ◇ tbl\_ticket.seatNr

## Meilenstein 2: Implementierung

Csv files von diversen Webseiten wie Mocharoo.com oder IMDB generiert und editiert.  
Im Java-Programm wurden dann nur noch Werte als Kreuzprodukte kombiniert und  
zusätzlich Shows, bzw. Reservierungen mittels Zufallsgeneratoren implementiert.