Project Phase 1 and Assumptions

Nicholas Kory

ID 862-18-9331

nkory003@ucr.edu

Bryan Parada

bpara001@ucr.edu

ID 861-02-7456

April 28, 2020CS 166 Database Management Systems

Instructor: P. Bakalov

Section 22

TA: Qizhong Mao

In completing this homework we consulted…

* Gehrke, Johannes, and Raghu Ramakrishnan. *Database management systems third edition*. McGraw-Hill, 2003.
* Bakalov Petko’s CS 166 Database Management Systems Video Lectures and Lecture Slides. University of California – Riverside.
* Qizhong Mao’s CS 166 Database Management Systems Video Lab, Lab Assignments & Solutions, and Lab Notes. University of California – Riverside.

No code was written for this assignment.

See next page for Project Phase 1 Assumptions.

Project Phase 1 Assumptions

Overall Assumptions

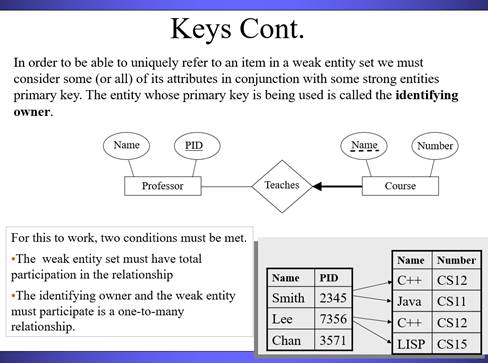
* We can create unique primary keys for tables (primary keys outside of the requirements, e.g., Cinema, Theater, Movie) to avoid weak entities.
* We avoided a ternary relationship between Seating, Booking, and Show by creating three binary relationships among the respective entities.
* Annotations are made next to certain document elements to explain our reasoning (e.g., why this is bold, why this is an arrow).
* The UI/front end will make updates to multiple tables when a single Booking is created or updated.

2.1 Movie

* Movie has a unique id attribute to make this a strong entity.

2.2 City

* States and Cities after modelled after the slide below. The state column in the State table are primary keys – they are unique and there is only one copy of each. The same cannot be said for Cities table – one zip may be used with different cities and one city may be used with multiple zips. However we can create one-to-many (one state to many city/zip relationships) relationship called Addresses and force total participation for the weak entity Cities in table Addresses. This allows us to follow the spec that says, “A city can have no cinemas.” So Cities use a self-contained primary key, which we then extend to the table Addresses and the relationship between relationships CinemaAddresses.



2.3 User Accounts

* As the same email cannot be used in multiple accounts, we use this as the primary key for the User table.

2.4 Cinema

* Cinema has a unique id attribute to make this a strong entity.
* Multiple cinemas in one city may have the same name (cinemas in the same company, like AMC) – no further assumptions were enforced.

2.5 Cinema Theater

* Theater has a unique id attribute to make this a strong entity.

2.6 Cinema Seating

* Seating has a unique id attribute to make this a strong entity.

2.7 Show

* Show has a unique id attribute to make this a strong entity.
* Shows can be filtered by the theaterid attribute since a primary key is unique to each theater.

2.8 Booking

* Booking has a unique id attribute to make this a strong entity.

2.9 Show Seating

* This is created using a relationship between Show and Seating.
* **The booking for the show seating is determined by the union of ShowBooking and SeatingBooking using the ShowSeating column values (seatid and showid).**