

Northeastern University

QwikTix.ai

Final Report

Oliver Fishstein, Sheryl Deakin
Karan Marwah, Joseph Wood

Professor Nathaniel Derbinsky
CS 3200 Database Design Section 3
Spring 2018

April 18, 2018

Abstract

So much about the movie industry has changed in the last ten years. Movie platforms like Netflix provide thousands of titles users can watch instantaneously for a cheap monthly rate. On Demand and iTunes, rentals make watching whatever movies Netflix doesn't have easy for just \$5 each. MoviePass allows users to go to any of the thousands of theaters currently available and watch any movie for \$10 a month. Even theaters themselves have seen big upgrades as luxury theaters with food and drink service as well as reserved seats have increasingly taken over. With so many ways to watch movies even from the confines of your home, it would seem like there's nothing left to improve. But, movie-goers now face an even simpler problem than how or where to watch. They simply don't know what to watch. As new services that pop up seemingly every day make it easier to watch movies, none have made picking a movie any easier.

This system first addresses the problem of finding a movie to watch. Almost all of the processes go toward making this step easier. Right off the bat, the user is shown information about what top genre's other users are viewing as well as the top studios. This gives the user an idea of trends in the movie industry. The user can also choose to love specific movies. This allows the user a way to view specific movies later, and we'll even remind them of movies they've loved but have not yet ordered. Once the user orders this movie, the app will make a note of it and not tell them about it anymore. It's already easy to see that the system provides more and more benefits the longer a user uses it. Down the road, this information will become even more useful as machine learning algorithms could be introduced. If a user has a specific movie in mind, they can search for it, or maybe there's a director they like. Searching for the director will bring up every movie they've been apart of which will give the users a way to explore films they may not have seen before. Once a movie has been selected, it's easy to purchase a ticket to a theater.

Description

Problem Overview

The mission of QwikTix.ai is to use state-of-the-art AI methods to disrupt the movie tickets industry. The vision: Fandango meets IMDB meets Apple's TV app. In order to accomplish this, a database is needed that can support the following tasks with a database that supports the existence of Users, Movies, Credits, and Order :

- Register a new user
- Record that a user loved a new movie
- Order a ticket from a local movie theatre
- Credit an existing actress for a movie
- Provide a ranked list of revenue generated from the top-10 studios
- Find all movies directed by a person (supplied via last name)
- Load the cover images and names of movies ordered by a particular user
- Find all movies released this year that a user loves but has not ordered
- Find all people (name, picture, and role) credited for a particular movie (supplied by name)
- Provide a ranked list of revenue generated from the top-3 movie genres

Users provided information about themselves upon registration: first/last name, email address (used for login/notifications), a password, a profile picture, contact phone number(s), and a contact address (street, city, postal code, country). They have the ability to search for movies and love them in addition to ordering movie tickets or streaming access to movies.

Movies contain various production information such as a name, a release date, and some number of images. Movies are classified into genres and are produced by a single studio.

To make finding movies easier, movies have associated credits of people who were involved with the production of the movie. Each of the people credited has a first name, last name, picture, and date of birth and associated roles such as directors, producers, and actresses/actors. A person can have multiple roles, and for actresses/actors specifically, the character(s) played is indicated as well as have an associated picture.

The primary source of revenue for QwikTix.ai is users ordering via partner vendors which can be theatres or streaming services. Theatres have names and geographic locations while streaming services have a name and affiliate URL. If a user is interested in viewing a movie, the system queries vendor APIs for ticket/streaming access. A user order includes a dollar amount, date/time of the order, # tickets, seats, expiration of streaming, etc. for a particular movie from a particular vendor. An external payment processor is used, so no need for credit cards, just a confirmation number is maintained for the order.

The following entities are used to accomplish all of these tasks and store the relevant information:

Entity Descriptions

SiteUser

In order to keep track of the site's user, the system has SiteUsers. Each user's info consists of their first and last name, email, password, profile picture, and address. To allow for contacting users, we also relate to the entity, SiteUserPhone, which allows access to a list of contact phone numbers for a particular user. A user also relates to the LovedMovie entity to provide a listing of movies that user has indicated that they love and want associated with their account. In order to enable movie rentals or ticket purchases, a user also relates to a MovieOrder to allow for access to a list of orders a user has made and allow for tracking who made the order.

SiteUserPhone

The system has SiteUserPhones which allows for the matching of a SiteUser to a phone number and retrieve a list of contact phone numbers for each user.

LovedMovie

The system allows for user's to love a movie through LovedMovies. This allows the system to match a movie with a SiteUser who has indicated that they love a particular movie. This allows the system to get a listing of movies loved by a user or users who love a particular movie. This information can be used for further recommendations to users because it allows a user to show their taste preferences.

Movie

A core functionality of the system is storing movies and all their relevant production information. Each movie has its name, genre, release date, studio, and cover picture. Movies relate to MovieOrders by allowing a MovieOrder to reference a particular movie that was purchased. In order to retrieve a list of related pictures for the movie, it relates to a MoviePictures in order get any photo that has been matched with that movie. Lastly, it relates to a Credit so that Professionals who worked on the production for the movie can be credited for their role.

MovieOrder

A MovieOrder consists of a confirmation number, order time, and dollar amount for cost along with references to the vendor purchased from, the user who purchased the tickets/rental, and the movie that was purchased. The references allow for getting more specific info about the vendor from Vendor, the user from SiteUser, and the movie from Movie. A MovieOrder is one of two types: streaming or theatre, which gives more specific information about the type of order completed. Streaming orders information is stored in StreamingOrder, and theatre orders

information is stored in TheatreOrder. This information can be referenced through the order's confirmation number.

StreamingOrder

A StreamingOrder consists of a confirmation number to reference the main MovieOrder and an expiration for when the stream is no longer valid.

TheatreOrder

A TheatreOrder consists of a confirmation number to reference the main MovieOrder and a ticket count for the amount of seats purchased.

Vendor

A Vendor consists of a name and id. This provides information about a specific vendor that movies can be purchased from and can be referenced by a MovieOrder. By having it as a separate entity, it is easy to access any orders related to a specific vendor and reuse the information for many orders. A vendor can either be streaming or theatre whose additional information can be accessed from StreamingVendor or a TheatreVendor through the id to differentiate the necessary fields.

StreamingVendor

A StreamingVendor extends the information of a Vendor and has an URL with access information for the specific streaming service.

TheatreVendor

A TheatreVendor extends the information of a Vendor and provides information for the specific theatre location.

Professional

A Professional consists of a PersonID, FirstName, LastName, Picture, and a DOB. A Professional is any Hollywood actor, director, or other production role that could be credited in a movie. This allows for a professional to have many credits. It relates to a Credit through the PersonID.

Credit

A Credit consists of a role and references to a Professional who is being credited and the Movie that this credit is associated with. This allows someone involved in the production to be linked to a specific Movie. If the role is an actor/actress, then it is associated with a CreditCharacter to provide more detailed information about the credit through its id.

CreditCharacter

A CreditCharacter consists of a picture of the character, a name, and a reference to the Credit it applies to. This provides more detailed information about a Credit for actor/actress roles.

MoviePictures

A MoviePictures consists of a picture and a reference to the movie it is from. A MoviePictures allows for the creation of a listing of pictures related to a specific movie that can be accessed from the movie.

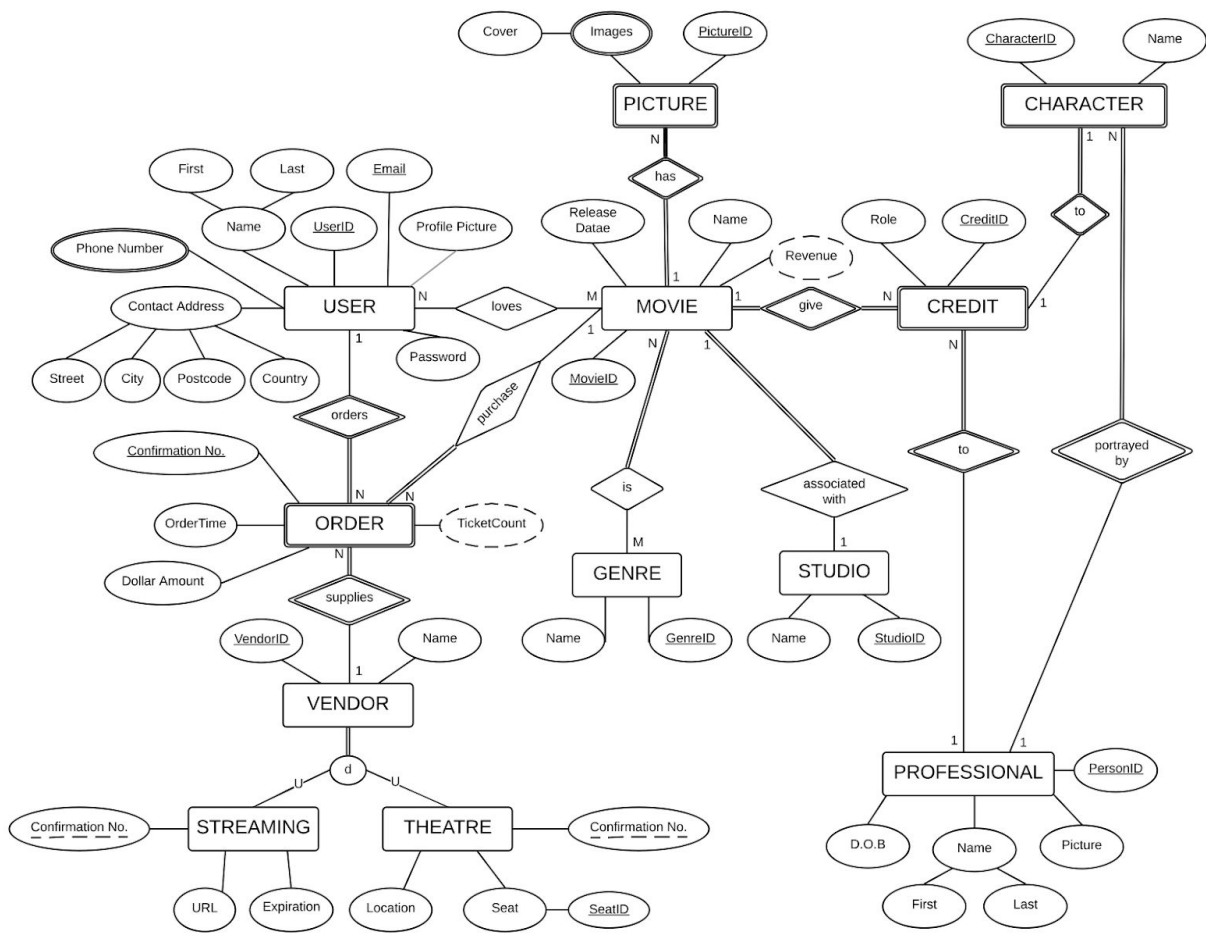
Studio

A Studio just consists of a name. This allows for a Movie to be credited as produced by a particular Studio. By creating this entity, it allows for an easy way to access the list of movies that have been produced by that studio.

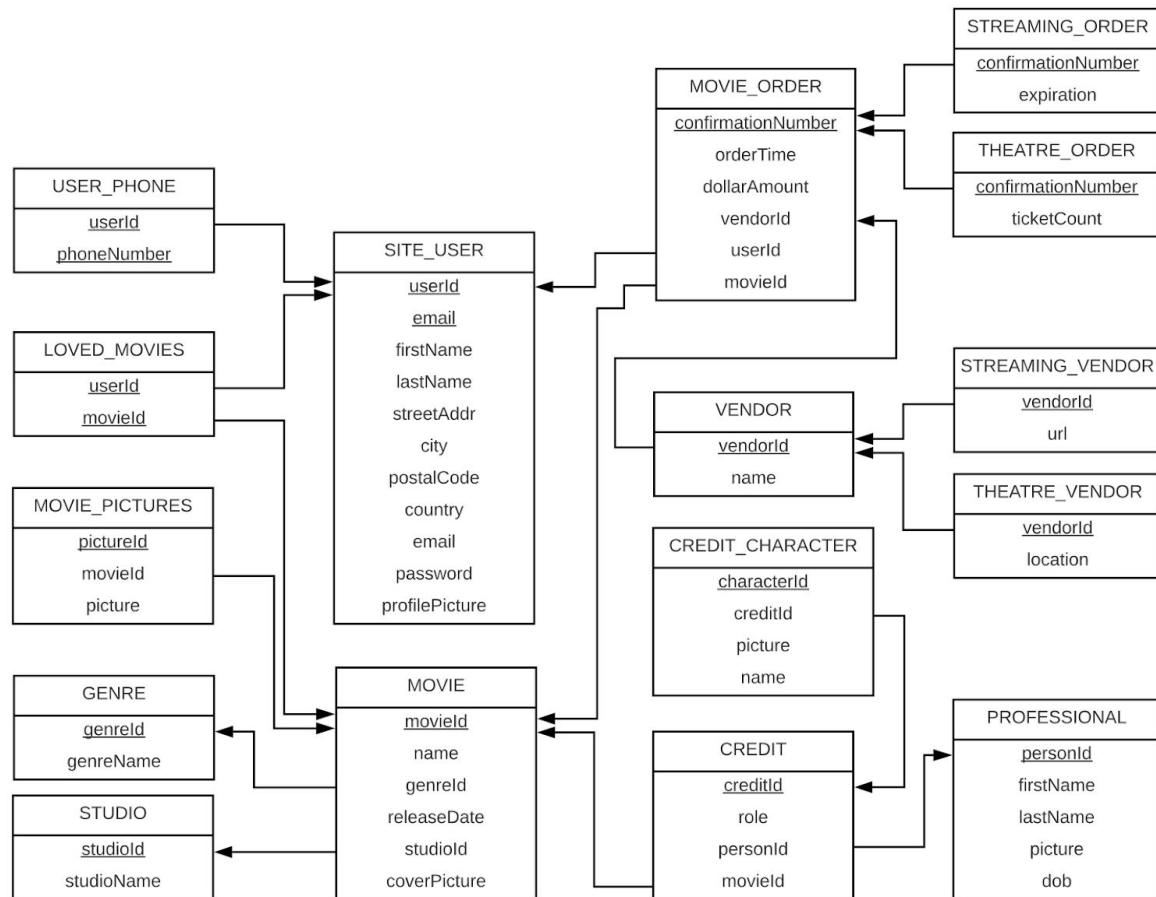
Genre

A Genre just consists of a name. This allows for a Movie to be associated with a particular genre. By creating this entity, it allows for an easy way to access the list of movies that have been associated with a genre.

ER Diagram



Normalized Relations



Physical Design

For our physical design, we used primary key indexes for all tables. This allowed us to create clustered tables and more convenient indexing and joining. Since any primary key involves creating a unique index when declared we ensure both logical consistency and performance. There are also no matching attributes across relations that are not either a foreign or primary key, to avoid producing invalid tuples in such relation. We also ensured to not combine attributes from multiple entity types and relationship types into a single relation. This would have resulted in semantic ambiguities that would have made some relations less client-friendly.

All of the weak entity types, like MovieOrder, CreditCharacter, and MoviePictures, create a corresponding relation that includes all simple attributes. CreditCharacter and MovieStudio, also have total participation and a binary 1-to-1 mapping. Every movie has a studio, and every studio has a movie. Every character has to have a credit, however, not every credit is a CharacterCredit and can belong to a Professional.

We used foreign keys for multivalued attributes, like MoviePictures and UserPhone, since multivalued attributes cannot work as primary keys and directly violates First Normal Form (1NF). Thus, we created new relations for those attributes with the corresponding primary keys as foreign keys.

For disjointed specialization of the type of vendor, we chose to have multiple relations with subclasses only. Foreign keys are used by subtype entities to enforce the dependent relationship between StreamingOrder and TheatreOrder, as well as StreamingVendor and TheatreVendor.

We created a relationship entity for LovedMovies which consists of two foreign keys to the SiteUser and Movies to force the pairing between them. A user can love any number of movies, and a movie can have any number of users loving it. Because they have a binary M-to-N cardinality, the new relation LovedMovies and the combination of both the user and movie's primary key compose LovedMovies' primary key.

Screenshots

Register User

≡ Register

First Name	Last Name	Address	City	
Joe	Shmo	19 Peterborough Street	Boston	
Country	Zip Code	Email	Password	Pic URL
USA	02120	joe@google.com	blabla	asdfd

SUBMIT

≡ Login

Email

Joe

Password

.....

Registration successful, please sign in

REGISTER

SIGN IN


Love Movie
Order Movie
Credits For a Movie

Movie Name

Search

Titanic

SEARCH



Titanic

1997

Credits

Role	First Name	Last Name
Matte Painter	Brigid	Greaney

LOVE

ORDER


Search For Movies By Last Name of Director

Director Name

Search

Williams

SEARCH



Lion King

1994

Credits

Role	First Name	Last Name
Director	Ann	Williams

LOVE

ORDER


Top 3 Genres (Revenue)


Top 3 Genres (Revenue)	
Genre Name	Revenue
Action	116
Comedy	98
Drama	62

A Users Ordered Movies (Pics and Title)

≡ Orders

Ordered Movies

Titanic
1997

The Incredibles
2004

Credit existing actor/actress for a movie role

≡ Credit an existing actor/actress for a role in a movie

Name

Joseph Wood

Title

Titanic

Role

Actor

CREDIT

Name

Joseph Wood

Title

Titanic

Role

Actor

CREDIT

Successfully Credited

Top 10 Studios Revenue

Top Studios (Revenue)

Studio Name	Revenue
Universal	117
Pixar	116
Warner Bros.	58
Paramount	54
IFC	40
Viacom	40
Sony	40
Columbia	40
Fox	13
Disney	6

Loved But Not Ordered Movies Released this Year

Loved Movies You Havent Ordered This Year



The Dark Knight
2008



Black Panther
2018

Project Retrospective

The most interesting part of this project was the real world applications of the database and the queries that we were asked to develop. So many people access media online on a daily basis to watch TV or movies so it was very interesting to work on developing a database to store all the relevant information that a user would want to access about a movie in order to find what to watch or keep track of movies they already have watched in addition to all of the information a website needs to have to facilitate movie ticket sales and rentals. This enabled us to get a very basic understanding of the building blocks necessary for services like Netflix, Fandango, or Amazon Video to exist.

The thing that we disliked the most was splitting the work amongst the group due to how dependent different aspects of the project were on each other. We found it difficult to split each individual code deliverable and the diagrams even when using collaboration tools so, it was hard for group members to get exposed to every aspect of the project after the initial discussion on how to design our system. This led to some waiting for different aspects to be developed before proceeding further in the project like waiting for DDL and DML to be completed until queries could be developed.

The easiest part of this project was deciding how to structure the database because from the project narrative and our understanding of how similar systems work already it was pretty intuitive. It was pretty quick to reach consensus on how our entities would look and what information would be required for the system. With the prior experience our group had developing systems in the past, this project was overall manageable for us.

The hardest part had a lot to do with what we learned from working on a large-scale project. We found communicating about different aspects of the project hard at first because it was easy for specific information to get lost in communication which led to some discrepancies in our milestone work. Going forward, we worked a lot on our team communication to ensure that the different elements of our project would be of a high quality and agree with each other. This also made dividing work and helping with bugs an easier process. The most important part of large-scale projects was definitely keeping in communication and keeping to a schedule with clearly assigned tasks in order to stay on track and accomplish everything.

Conclusion

This system makes it easy for a user to pick a movie and organizes the data in such a way that further work will easily build off this. Users are given numerous options and metrics for deciding a movie as well as the ability to browse through all the choices. Once a movie is selected, the users can purchase tickets to movies directly on the site to minimize any extra steps. This work also laid the foundation for future projects, namely a machine learning project. Using the information this system saves in the database about what users see and what users like it, creating a machine learning project would be simple as all the data is already easily accessible. While not there yet, this system is well on it's way to disrupting the movie business by aiding in the selection process.