Quicktix.ai

Final Presentation

Oliver Fishstein, Sheryl Deakin, Joseph Wood, Karan Marwah
CS 3200 - Database Design, Section 3 - Spring 2018

Project Motivation

Big Data Trends

- Custom data solutions to track key metics and spot operational inefficiencies
- Predictive Analytics
- The vision: Fandango meets IMBD meets Apple's TV app
- The mission: Use state-of-the-at AI methods to disrupt the movie tickets industry.
- "What movie should I watching evening?"
- Movies have predefined set of movies similar to them
    - User's have a predefined set of "previously seen, liked, movies"

System Description

System

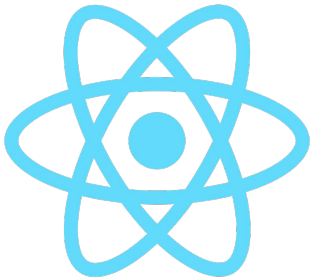
- Creates reports/metrics that makes picking easier
- Browsing
  - Search for movies
  - View what's popular
- Liking movies
  - Learning
  - Save for later
- Order a ticket

Tasks

- Register a new user
- Log in to an existing account
- Credit a professional for a role
- View a list of genres by revenue
- View a list of studios by revenue
- View all movies loved but not ordered by a particular user
- Order a movie ticket to a theater
- View a user’s past movie orders
- Search for movies by name or director’s last name
- Generate all the reports given a directors name and a year

System Architecture

The Breakdown



- React/Redux Frontend for modular components and a transparent data flow.
- Node/Express Backend for easy to create endpoints and relatively small setup time
- MySQL for DBMS due to the ACID principles

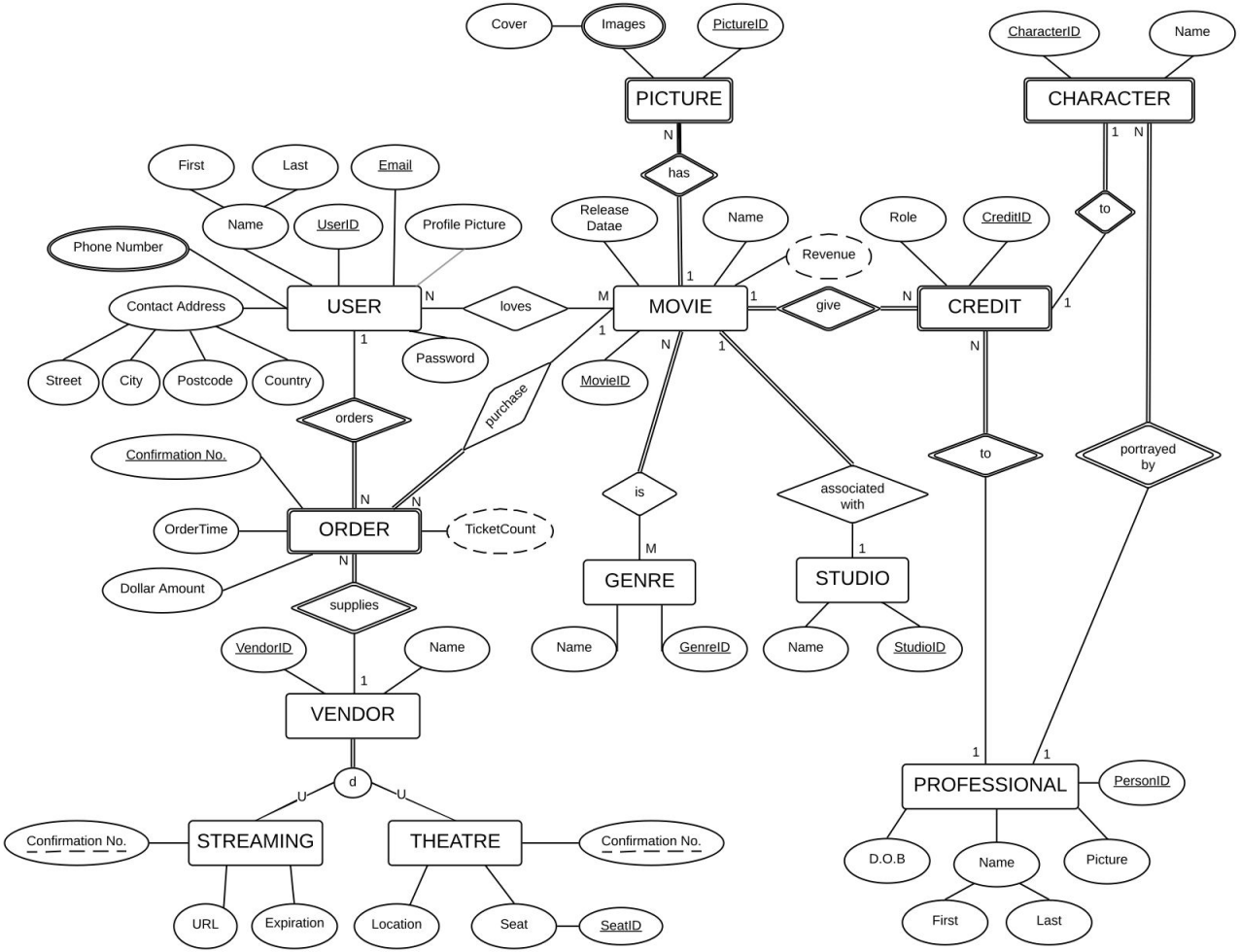
How does it all talk?

- For simplicity's sake everything is hosted locally (prototype)
- Express App runs on localhost port 3000 and hosts our endpoints
- React App runs on localhost port 3001 and makes fetch request to port 3000
- MySql runs in parallel on the system and connects to Express via the mySql library and providing your username and pw for the database

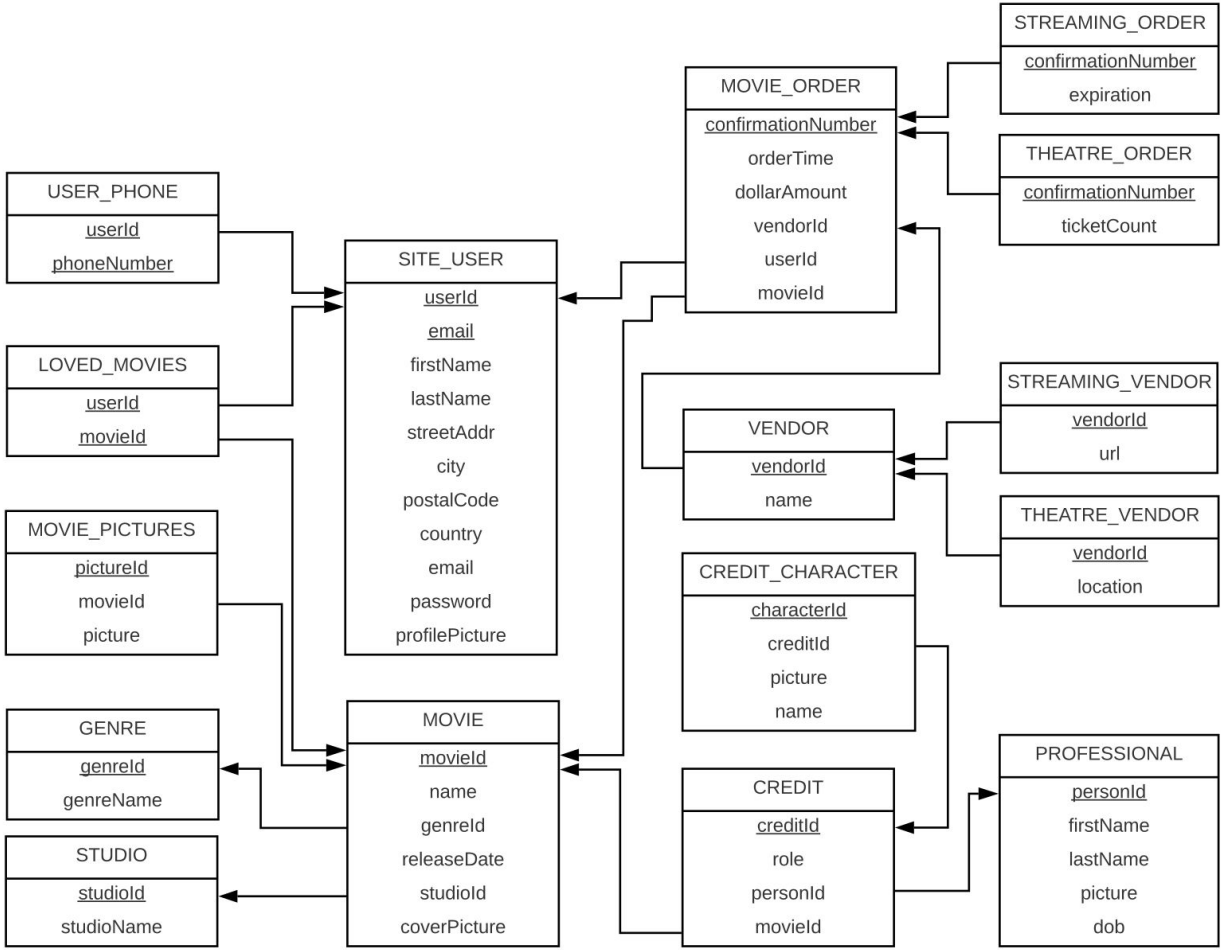
How to run

- Clone the dbdesignproject directory and download the ddl and dml for our DBMS
- Run MySQL and create a database with our ddl and dml as source
- In the dbdesignproject directory, edit the username, password, and database name in the server.js file to match your system.
- Run npm install in the dbdesignproject directory to get all depenencies
- Brew install ‘yarn’ if you don’t have it
- Navigate to dbdesignproject>src. Enter the command ‘node server.js’ (starts backend)
- Navigate back up to dbdesignproject. Enter the command ‘yarn start’ (starts frontend)
- Go to localhost:3001 in your browser and you’re ready to roll!

Database Design



Entity Relationship Diagram



Relational Model

Physical Design

- Decided to use primary key indexes for all tables
  - Easy indexing and joins
- Foreign keys used by subtype entities to enforce dependent relationship
  - StreamingOrder and TheatreOrder
  - StreamingVendor and TheatreVendor
- Foreign keys used for multivalued attributes like MoviePictures and UserPhone
- Relationship entity for LovedMovies consists of two foreign keys to force pairing of Movies to SiteUsers

Data Sources

- Wikipedia and Google Images
  - Professional, Movie, MoviePictures, Credit, CreditCharacter
- Popular Lists
  - Genre, Studio, Vendor, StreamingVendor, TheatreVendor
- Generated By Team
  - SiteUser, SiteUserPhone, LovedMovies, MovieOrder, StreamingOrder, and TheatreOrder

Thanks!