Group 25 (Konami Code) - Project Step 3 Draft

Team members: Jesseline Velazquez, Anthony L Clary
Project Title: IncrediFilms, a Cinema Experience
Website: http://flip1.engr.oregonstate.edu:40593/

Step 2 Feedback from Peer Reviewers

Reviewer 1

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**

  Yes, the model follows the database outline and the ERD

- **Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

  a) Yes, there is consistency amongst the naming conventions between overview, outline, & ER.

  b) Although entities are not plural, I'm sure there was a reason for it since it made it through step 1 review. As for the attributes, they are singular.

  c) Yes, names are capitalized where appropriate.

- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**

  Schema is easy to read. In fact, the whole document is so well organized that it made everything easy to read!!!

- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**

  Yes

- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**

  No

- **Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**

  Yes. Also contains appropriate comments and even a subquery.

- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**

Yes

- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**

  The SQL file and Schema are a match. Cascades are included prior to table creation at the DROP TABLE line.

- **In the SQL, are relationship tables present when compared to the ERD/Schema?**

  Yes

- **In the SQL, is all example data shown in the PDF INSERTED?**

  Yes

- **Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?**

  File is well structured (hand authored) with comments and sectioned based on Entity. For ex: Entity movie would have its table created and data inserted below. On the other hand, I had all my tables created first then its values inserted after.

  Overall, was aesthetically pleasing to look at and very organized (esp the first page).

Reviewer 2

Very nice project concept, and excellent formatting for both the PDF and SQL files. I've bolded the important feedback below. Being from Chicago, I especially liked the examples you picked for theater locations!

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - **Yes.** The schema exactly matches the database outline and ERD.
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - **Mostly:**
    - a) The naming between the overview, outline, ERD, and schema is consistent for both entities and attributes.
    - b) **Entities are singular instead of plural as expected.** Attributes are singular as expected.

    - c) Capitalization is consistent. All attributes are entirely lowercase, while all words within entity names are capitalized. All words are separated by underscores.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - **Yes.** The tables are appropriately spaced and no relationship lines are crossed.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - **Yes.** Both the "Showtime" and "Movie_Genre" intersection tables contain the appropriate foreign keys and enable an M:N relationship (between theaters and movies, and movies and genres, respectively).
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - **Somewhat.** Ticket.payment_method is redundant data that could be prone to inconsistent entry (e.g. "Credit" vs. "Credit card" vs. "Credit Card"). It could also cause several types of anomalies (e.g. no indication that cash is accepted until someone pays cash, deleting all cash Ticket entries obscures the fact that cash is accepted). **I'd recommend creating a separate category table for payment types.** Movie.mpa_rating is similar but probably less problematic because those values come from an authoritative third party and are less likely to change over time. I did not notice any partial or transitive dependencies.
- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!
    - **Yes.** The SQL file imports to phpMyAdmin without any errors.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - **Yes.** It was an especially good choice to use DATETIME instead of TIMESTAMP for Showtime.showtime_date_time, since the focus in on calendar/clock times instead of the exact Unix time when a showtime occurred.
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - **Mostly.** All primary and foreign keys are correctly defined. All of the DROP TABLE statements use ON DELETE CASCADE. However, I think the intention is that the CREATE TABLE statements should also have ON DELETE behavior defined (what happens if a genre is deleted after table creation?), so **I'd recommend adding that in as well**.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - **Yes.** The Showtime and Movie_Genre intersection tables are both present, and contain the attributes and keys described in the ERD and schema.

- In the SQL, is all example data shown in the PDF INSERTED?
    - **Yes**, all example data from the PDF is inserted in the SQL file and imports with the expected values.
- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
    - **Yes.** The SQL file is well-commented, contains citations where data came from outside sources, and is structured so as to be very readable.

Reviewer 3

Group 25 Project Step 2 Review - Review by James Adelhelm

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

- Yes, there is consistency in matching.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- a) Yes, there is consistency between the overview, outline, ER, and schema.
- b) Entities are singular instead of plural. Attributes are singular. RECOMMENDATION: I'd recommend updating to meet requirements.

- c) Capitalization is consistent. Attributes are lowercase and words in the entity names are capitalized. Underscores are used to separate words.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

- Yes, the schema is easy to read as the tables are appropriately spaced and lines are not crossed.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

- Yes, the intersection tables are properly formatted. The "Showtime" and "Movie_Genre" intersection tables have the appropriate FKs and there exists a M:N relationship.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

- No, it does not suggest non-normalized issues. RECOMMENDATION: You could potentially create a category table for types of payment in order to avoid inconsistent entries and redundant data input.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

- Yes, the SQL file works.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

- Yes, the data types are appropriate. Like Joshua said, good idea to use DATETIME for Showtime.showtime_date_time.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

- All the PK's and FK's are defined correctly. The appropriate CASCADE operations are declared for the DROP TABLE statements.

- Recommendation: consider adding ON DELETE behavior defined for the CREATE TABLE statements.

In the SQL, are relationship tables present when compared to the ERD/Schema?

- Yes, the Movie_Genre and Showtime intersection tables are both present. They contain the attributes and keys described in both the Schema and ERD.

In the SQL, is all example data shown in the PDF INSERTED?

- Yes, the data is inserted with the expected values.

Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

- Yes, the SQL is hand authored and well commented.

Reviewer 4

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes, the ERD and DB Outline match the schema.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

a) Consistency between the overview, outline, ER, and schema is present.

b) Entities are singular instead of plural. Attributes are singular.

c) Consistent capitalization. Attributes are lowercase and words in the entity names are capitalized. Underscores are used where appropriate.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, intersection table are properly formatted. The "Showtime" and "Movie_Genre" intersection tables have the FKs and a M:N relationship.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

No, it does not.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

Yes, the SQL file works.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Data types are appropriate.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

All the PK's and FK's are defined correctly. The appropriate CASCADE operations are declared for the DROP TABLE statements.

In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes.

In the SQL, is all example data shown in the PDF INSERTED?

Yes

Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

Yes

Step 2 Summary of Feedback

Schema Feedback

- Multiple peer reviewers recommended updating entity names to plural.

DDL Feedback

- Create a category table for types of payment in order to avoid inconsistent entries and redundant data input (e.g. "Credit" vs. "Credit card" vs. "Credit Card"). This could also cause several types of anomalies (e.g. no indication that cash is accepted until someone pays cash, deleting all cash Ticket entries obscures the fact that cash is accepted).
- Similarly, Movie.mpa_rating may cause issues but less so because those values come from an authoritative third party and are less likely to change over time.
- All of the DROP TABLE statements use ON DELETE CASCADE. However, I think the intention is that the CREATE TABLE statements should also have ON DELETE behavior defined (what happens if a genre is deleted after table creation?), so I'd recommend adding that in as well.
- A peer reviewed Recommendation: consider adding ON DELETE behavior defined for the CREATE TABLE statements.

Fixes based on Feedback from Step 2

- No change to the tense of entity names as we are following the book’s format.
- No change to the payment methods in the Ticket table as suggested by a reviewer. Instead this will be handled by the front end as a drop down menu in the “Create New Ticket” form.
- Similarly, no change to Movie.mpa_rating as this will be handled in the UI.
- The DROP TABLE statements are intended to drop the given table if it exists BEFORE creating that table again. The CREATE OR REPLACE TABLE statement does not currently play nice with MySQLWorkbench.
- The CASCADE command has been removed from the DROP TABLE statements.
- ON DELETE CASCADE constraint added to FOREIGN KEY constraint definitions under the Movie_Genre table. The justification is that if the parents are deleted (i.e. Movie or Genre record), we no longer need its association recorded in the Movie_Genre table.
- No delete condition constraints added to the Showtime or Tickets tables. Justification is that these tables should act as indelible records when possible even if a movie, showtime, or customer has since been deleted from other tables. NOTE: It will still be

possible to manually delete or edit records in these tables via our UPDATE and DELETE Queries.

Upgrades since the Step 2 Draft

- Foreign Key constraints assigned aliases for easy manipulation, if needed.
- Added CHECK constraints for both Movie.mpa_rating and Ticket.payment_method as we will likely utilize hard-coded lists for input selection of these values on the frontend (note: no more than 5 possible responses).
- Removed the NOT NULL constraint on the customer_id foreign key attribute in the Ticket table.

Step 1 Feedback from Peer Reviewers

Reviewer 1

I "lol'd" at the "no $12 popcorn here...

Great idea to use for the db.

- Does the overview describe what problem is to be solved by a website with DB back end?
  - Yes. This overview describes the various entities the group will have to keep track of in the database.
- Does the overview list specific facts?
  - Yes, there are specific facts related to the problem statement and what they are trying to solve.
- Are at least four entities described, and does each one represent a single idea to be stored as a list?
  - Yes, this group lists 5.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
  - Yes, very clearly.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
  - Yes to all.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
  - One thing that I might mention here, is you might want to lower case the _ID portion of the column names to save you some typing time and consistency. I know some settings in MYSQL can be case-sensitive so just a thought.

Reviewer 2

- Does the overview describe what problem is to be solved by a website with DB back end?
  - The over view gives a broad view of the organization and it is clear that a DB back end would be helpful in managing such an organizations.
- Does the overview list specific facts?
  - The overview has some specific facts like the cost of popcorn, and the organizations dedication to paying employees a living wage, but it does not list the number of patrons, employees, movie theaters or volume of goods sold. These might be helpful numbers when designing and implementing a data base that fits the needs of IncrediFilms
- Are at least four entities described and does each one represent a single idea to be stored a s a list?
  - There are a total of 6 Entities each with their own single idea to be stored in a link. At first I thought showtimes could be an attribute of movies, but upon further inspection, the current Entity makes a lot of sense.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

    - Yes, the outline is very thorough in describing the datatypes, relationships and constraints of the entities and attributes
    - The relationship between showtime, auditorium and movie threw me for a bit of a loop at first, but I think the way you have it formatted makes sense. Is it that each showtime will have a FK of the movie and auditorium so that they can be differentiated from the same movie being shown at the same time in a different auditorium? just an opinion, but "showings" might be a better title for that entity
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
    - Yes, they have correctly formatted 1:M relationships and They have a M:M relationship. I'm a little confused with the ERD as it doesn't list all of the same attributes and FKs as the outline. For instance, the outline of the ticket entity mentions an auditorium FK, but the ERD diagram doesn't show that relationship. There is a footnote explaining some sort of inheritance, but that isn't clear in the diagram. Maybe the tickets should have a M:1 relationship with the auditorium? Either that, or the outline of tickets should only be linked to the showing which includes the auditorium, and the ticket should make no direct reference to auditorium.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - a) it looks like the diagram and the outline are consistent in naming
    - b) Entities are all plural, and attributes are singular
    - c) I noticed that under showtimes in the ERD, the movie_id FK is capitalized and that under sales, the customer_id FK is also capitalized. Otherwise good

Nice work! Your team seems to have taken on a complex schema with this project and done a good job of it!. wishing you all the best!

Reviewer 3

The overview is detailed and gives a great understanding of what the website and back end will look like. There are specific details in the overview missing, as the cost of the popcorn seems to be a joke to hook the reader in rather than fact. I think there should be more specificity and hard numbers to get a clearer understanding and picture of the scale of the project.

There are more than four entities, with six entities representing individual ideas to be stored as lists. All of the entities make logical sense.

The outline is well-thought out and describes all the datatypes, relationships, and constraints. I really appreciated the bolding in the outline, as it clearly showed which entities were the focus.

The ERD is put together well and describes the DB similarly to the outline. All of the 1:M and single M:M relationship seem to be in order as well.

Thankfully the diagram and outline are consistent almost all the way throughout, though there were a couple inconsistencies, mainly with how ID is capitalized in the outline but not in the diagram. I think if your team is going to work with snake case for the attributes, then keep ID lowercase (e.g. movie_id).

Great job!

Reviewer 4

- Does the overview describe what problem is to be solved by a website with DB back end?
  - Yes! You did a great job of describing the problem and showing why a website with a DB back end is necessary.
- Does the overview list specific facts?
  - It does! I would say it maybe needs another number somewhere but I thought it was very detailed.
- Are at least four entities described and does each one represent a single idea to be stored a s a list?
  - Yes, the plan describes 6 different entities described as lists.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
  - The outline of entity details describe the purpose of each and list datatypes and constraints really well. The outline also describes the relationships between entities, each of them having multiple relationships working together.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
  - As far as I can see the 1:M relationships are correctly formulated. There is at least 1 M:N relationship, I counted two. The ERD presents a very logical view of the database.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
  - For the most part, everything looks pretty consistent besides some capitalization issues, specifically between the ID portion on the outline and the diagram.

Great job!!

Reviewer 5

- Does the overview describe what problem is to be solved by a website with DB back end?

The overview describes that IncrediFilms is a movie theater company that seeks to combine a nostalgic atmosphere with modern technology and logistics.

- Does the overview list specific facts?

Yes, the overview lists specific facts such as,

- IncrediFilms is a movie theater company that seeks to combine a nostalgic atmosphere with modern technology and logistics
- The company places emphasis on keeping costs low for consumers and providing fair compensation for employees
- All employees receive a living wage, at minimum.
- IncrediFilms requires a web-enabled front-end and a well-structured and robust database to manage organizational data.
- The website aims to facilitate employee and patron needs and to orchestrate scheduling, stock, and operational needs across all locations and manage organizational data efficiently, organized, and intuitive across locations.
- Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, at least four entities are described: Movie, Auditorium, Customer, and Showtime. Each entity represents a single idea to be stored as a list, such as movie information, auditorium information, customer information, and showtime information. Each entity has a set of attributes that define the properties of that entity, and relationships are also defined between the entities to show how they are related to each other.

- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes, the outline of entity details describes the purpose of each entity, lists attribute datatypes and constraints, and describes relationships between entities. Each entity is described as a single idea to be stored as a list, and includes attributes such as primary keys and foreign keys to indicate relationships between different entities.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

The outline of entity details describes the purpose of each, lists attribute datatypes and constraints, and describes relationships between entities. The relationships are correctly formulated as 1:M and there is at least one M:M relationship. The ERD presents a logical view of the database.

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

The entities use the Pascal case. All attributes use the snake case.

Step 1 Summary of Feedback

Overview Feedback

- The overview “broadly” describes what problem is to be solved.
- There is a lack of metrics and specificity provided so the reviewer is unable to get a clear picture about the scale of our project. A suggestion from the reviewer is to add hard numbers to the outline.

Database Outline Feedback

- Reviewers agree that the outline describes the purpose, listed attribute data types and constraints, and describes relationships between entities. There is at least 1 M:N relationship. There is some confusion about the relationship between Movie, Showtime, and Auditorium (now Theater). A reviewer suggests a rename of “Showtime” to “Showings” for clarity.
- There is confusion about the relationship between the Ticket and the Auditorium (now Theater).
- There are some differences in the naming conventions, where some entities are plural but the attributes are singular.
- Some attributes are inconsistent with the capitalization. Reviewers suggest to lowercase the “_ID” portion of the attributes to save some typing time and consistency due to MySQL being case-sensitive.

ERD Feedback

- There are some inconsistencies in the entities’ that are present outline that no longer exist in the ERD and vice versa.
- The inheritance footnote does not clearly read in the ERD.
- Some attributes are inconsistent with the capitalization. Reviewers suggest to lowercase the “_ID” portion of the attributes to save some typing time and consistency due to MySQL being case-sensitive.

Fixes based on Feedback from Step 1

Overview Actions

- Metrics about IncrediFilms' movie-goer attendance and number of locations were added, showing an increase in both the number of theaters IncrediFilms operates and the growth in attendance in the same five-year period. This addition narrows the scope of our project and provides a clearer image to the reviewer as to why IncrediFilms needs to implement a database backend to manage their organizational data.

Database Outline Actions

- Separate orders and tickets/seats entities were collapsed into directly associating tickets with customers.
- The relationship between Movie and Showtime is 1:M, where one Movie can have multiple Showtime(s) at the theater. No changes were made to this relationship.
- The relationship between Auditorium (now Theater) and Showtime is updated to 1:M, where a theater can contain multiple showtimes, but each showtime can appear in at most one theater.
- No update was made to the name of the entity 'Showtime.' We felt the original is consistent with the naming convention of our competitors' apps.
- Updates to the overview to match the intended relationships shown in the ERD were made: namely, removing inactive attributes from earlier workings of the draft.
- Attribute names in the database outlines have been adjusted to all lower-case. This implementation was made to save time and for consistency going forward.
- Entity/table names updated to singular, capitalized names, following examples provided in the course textbook.
- Earlier footnote describing how the movie and/or auditorium associated with a ticket will be found has been removed. This will be handled by SQL queries and is beyond the scope of the current assignment.

ERD Actions

- Dropped an unnecessary entity (Sale).
- Added a new 'Genre' entity.
- Attribute names in the ERD tables have been adjusted to all lower-case. This implementation was made to save time and for consistency going forward.
- Updates to the ERD to match the intended relationships outlined were made: namely, adding FKs to match the relationships mentioned or removing attributes no longer included in the outline.

Upgrades to the Step 1 Draft Version

- Finalized Ticket/Seats entity name to ‘Ticket’.
- Updated the relationship between Auditorium (now Theater) and Showtime to 1:M, where one Theater can have zero or more Showtime, but only only Showtime per Theater.
- Added an entity, Genre, which will now hold a M:N relationship with Movie.
- Condensed Ticket/Seat and Order entities to one entity, Ticket.
- Added price and payment_method attribute to Ticket to account for loss of Sale entity.
- Updated the relationship between Customer and Ticket to 1:M instead of the previous 1:M Customer had with Sale.
- Renamed Auditorium entity to Theater and renamed attribute “auditorium_name” to “theater_name”.
- Added attribute “movie_year” to Movie entity.
- Removed not NULL constraint on the email attribute in the Customer table.

Upgrades since the Step 1 Final Version

- Overview description update to better describe IncrediFilm’s objects for customer and movie details data keeping.
- Reorganized the outline layout to coincide with the DDL.
- Updated price data type from decimal(3,2) to decimal(5,2) to account for four- and five- digit prices.
- Removed UNIQUE constraint from the email attribute in the Customer table.
- Added NOT NULL constraint to payment_method attribute in the Ticket table.
- Added NOT NULL constraint to theater_name attribute in the Theater table.
- Added NOT NULL constraint to foreign keys in the Showtime table.
- Added NOT NULL constraint to foreign keys in the Ticket table.
- Removed width (e.g. int(4)) from multiple int attributes. MySQL is depreciating the need for small int widths.
- Edited the ERD diagram to also include Movie_Genre intersection table, matching Schema diagram.
- Added language in the Database outline entities to clarify the foreign key relationships.

Project Proposal

Overview

IncrediFilms™ is a burgeoning player in the world of movie-goer experience. IncrediFilms theaters seek to celebrate a fusion between the nostalgic atmosphere of the red-velvet curtain theater and the web native systems that today's audience expects. Each IncrediFilms theater remains community-owned and operated, while utilizing a modern logistics backend—orchestrating scheduling, stock, and operational needs across their 12 locations. IncrediFilms prides itself in keeping costs to consumers low—no $12 popcorn, here—and employee compensation fair. The organization is proud to state that all employees receive a living wage, at minimum.

In order to accomplish each of their defining qualities, the organization relies on a modern tech backend that keeps operations efficient, organized, and intuitive across all locations. The number of IncrediFilms cinema screens statewide increased 20% between 2019 and 2023 to 12, with overall attendance growing more than 40% in the same time frame to a record 4 million unique movie-goers and counting (as reported in their earnings report.) With plans to open another two locations in the next two years, they can no longer rely on their fleet of Etch-A-Sketches alone to handle record keeping. To this end, IncrediFilms requires a web-enabled frontend that facilitates employee and patron needs. Underlying this frontend and other interface systems, a well-structured and robust database is crucial for managing organizational data.

Phase I of this database implementation is to streamline Movie scheduling amongst IncrediFilms' flagship Theaters and track financial performance of each Movie and Movie Genre via Tickets sold to Customers to decide Showtimes for the following week. This system must be capable of recording sales for 5,000 Showtimes in its 12 Theaters annually. Additionally, in acting as a record of IncrediFilm's screening history, the database should track basic movie details such as year, runtime, and its genre categorizations. These details will help to inform the online information system and ticket-buying experience for customers. Finally, IncrediFilms seeks to manage its loyal customer fanbase. The system will also need to keep track of IncrediFilms customer records, supplying contact information and linking customers with their ticket purchases. It is hoped that the data gleaned from these records will inform IncrediFilms decisions and strategy in the future, catering the experience to the organization's most prolific customers.

Database Outline in Words

1. **Entity 1: Customer** - records and stores information about each customer and/or potential customer
    - customer_id int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - first_name varchar(30) NOT NULL
    - last_name varchar(30) NOT NULL
    - dob date NOT NULL
    - email varchar(254)

    Relationship(s):
    - **1:M** relationship between **Customer** and **Ticket**; a Customer can be associated with zero or more Tickets, whereas a Ticket will be associated with one and only one Customer.

2. **Entity 2: Genre** - records and stores information about possible genres associated with movies.
    - genre_id int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - genre_name varchar(45) NOT NULL

    Relationship(s):
    - **M:N** relationship between **Genre** and **Movie**. Each Genre can have zero or more Movies associated with it; each Movie will have one or more Genres associated with it.
        - An intersection table, **Movie_Genre**, will facilitate this **M:N** relationship between Movie and Genre.

3. **Entity 3: Movie -** records and stores information of each movie IncrediFilms shows in their theaters.
    - movie_id - int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - movie_name varchar(100) NOT NULL
    - runtime_min int NOT NULL
    - mpa_rating varchar(5) NOT NULL
    - movie_year year NOT NULL

    Relationship(s):
    - **1:M** relationship between **Movie** and **Showtime**; each Movie can have zero or more Showtimes associated with it, each Showtime will be associated with one and only one movie.
    - **M:N** relationship between **Movie** and **Genre**. Each Movie will have one or more Genres associated with it; each Genre can have zero to many associated Movies.
        - An intersection table, **Movie_Genre**, will facilitate this **M:N** relationship between Movie and Genre.

4. **Entity 4: Showtime** - records and stores information about each showtime.
    - showtime_id int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - show_date_time datetime NOT NULL
    - movie_id int NOT NULL, FK
    - theater_id int NOT NULL, FK

    Relationship(s):
        - **1:M** relationship between **Showtime** and **Ticket**; a Showtime will be associated with zero or more Tickets, while a Ticket will be associated with one and only one Showtime.
        - **1:M** relationship between **Movie** and **Showtime**; a given Showtime will be associated with one and only one Movie; a given Movie can have zero or more Showtimes. movie_id is the foreign key in the Showtime table that facilitates the relationship between the two entities.
        - **1:M** relationship between **Showtime** and **Theater**; a given Showtime will take place in one and only one Theater; a Theater can have zero to many Showtimes. theater_id is the foreign key in the Showtime table that facilitates the relationship between the two entities.

5. **Entity 5: Theater** - records and stores information about each IncrediFilms theater location.
    - theater_id int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - theater_name varchar(50) NOT NULL
    - no_of_seats int NOT NULL

    Relationship(s):
        - **1:M** relationship between **Theater** and **Showtime**; a Theater can have zero to many Showtime instances. A given showtime will have one and only one associated Theater.

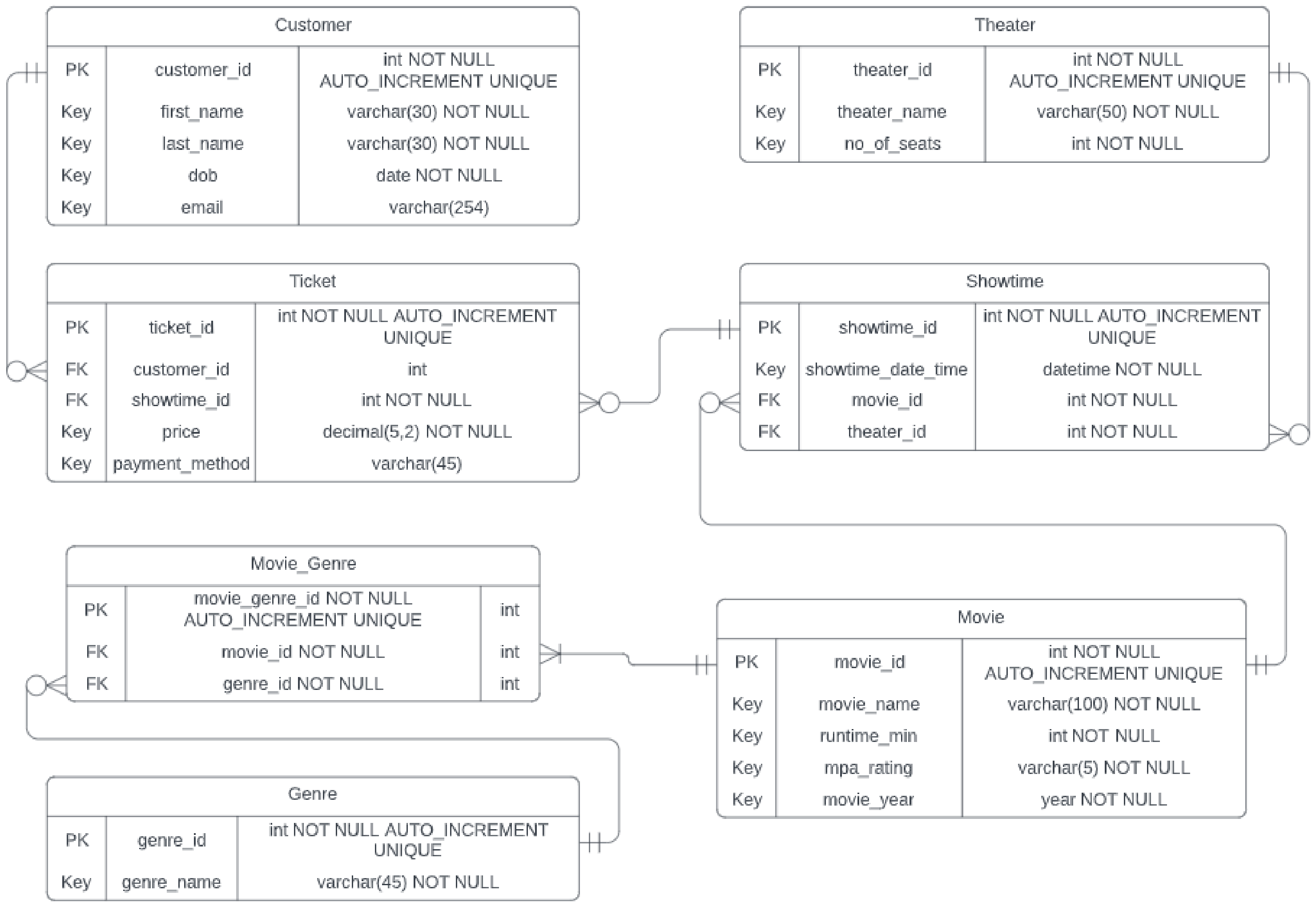
6. **Entity 6: Ticket** - records and stores the details of each movie ticket.
    - ticket_id int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - customer_id int, FK
    - price decimal(5,2) NOT NULL
    - payment_method varchar(45)
    - showtime_id int NOT NULL, FK

    Relationship(s):
        - **1:M** relationship between **Customer** and **Ticket**; a given Customer can have zero or more Tickets, while a given Ticket can have zero or one customer. customer_id is the foreign key in the Ticket table that facilitates the relationship between the two entities.
        - **1:M** relationship between **Showtime** and **Ticket**; a given Ticket will have one and exactly one Showtime, while a given Showtime can be associated with zero to many Tickets. showtime_id is the foreign key in the Ticket table that facilitates the relationship between the two entities.

7. **Intersection Table: Movie_Genre** - Facilitates the M:N relationship between Movie and Genre entities.
    - movie_genre_id int NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY
    - movie_id int NOT NULL, FK
    - genre_id int NOT NULL, FK

    Relationship(s):
      - Facilitates **M:N** relationship between **Movie** and **Genre**. Each Movie will have one or more Genres associated with it; each Genre can have zero to many associated Movies.
      - **1:M** relationship between Genre and Movie_Genre implemented with genre_id as the foreign key in Movie_Genre.
      - **1:M** relationship between Movie and Movie_Genre implemented with movie_id as the foreign key in Movie_Genre.

Entity-Relationship Diagram (ERD)



Customer		
PK	customer_id	int NOT NULL AUTO_INCREMENT UNIQUE
Key	first_name	varchar(30) NOT NULL
Key	last_name	varchar(30) NOT NULL
Key	dob	date NOT NULL
Key	email	varchar(254)

Theater		
PK	theater_id	int NOT NULL AUTO_INCREMENT UNIQUE
Key	theater_name	varchar(50) NOT NULL
Key	no_of_seats	int NOT NULL

Ticket		
PK	ticket_id	int NOT NULL AUTO_INCREMENT UNIQUE
FK	customer_id	int
FK	showtime_id	int NOT NULL
Key	price	decimal(5,2) NOT NULL
Key	payment_method	varchar(45)

Showtime		
PK	showtime_id	int NOT NULL AUTO_INCREMENT UNIQUE
Key	showtime_date_time	datetime NOT NULL
FK	movie_id	int NOT NULL
FK	theater_id	int NOT NULL

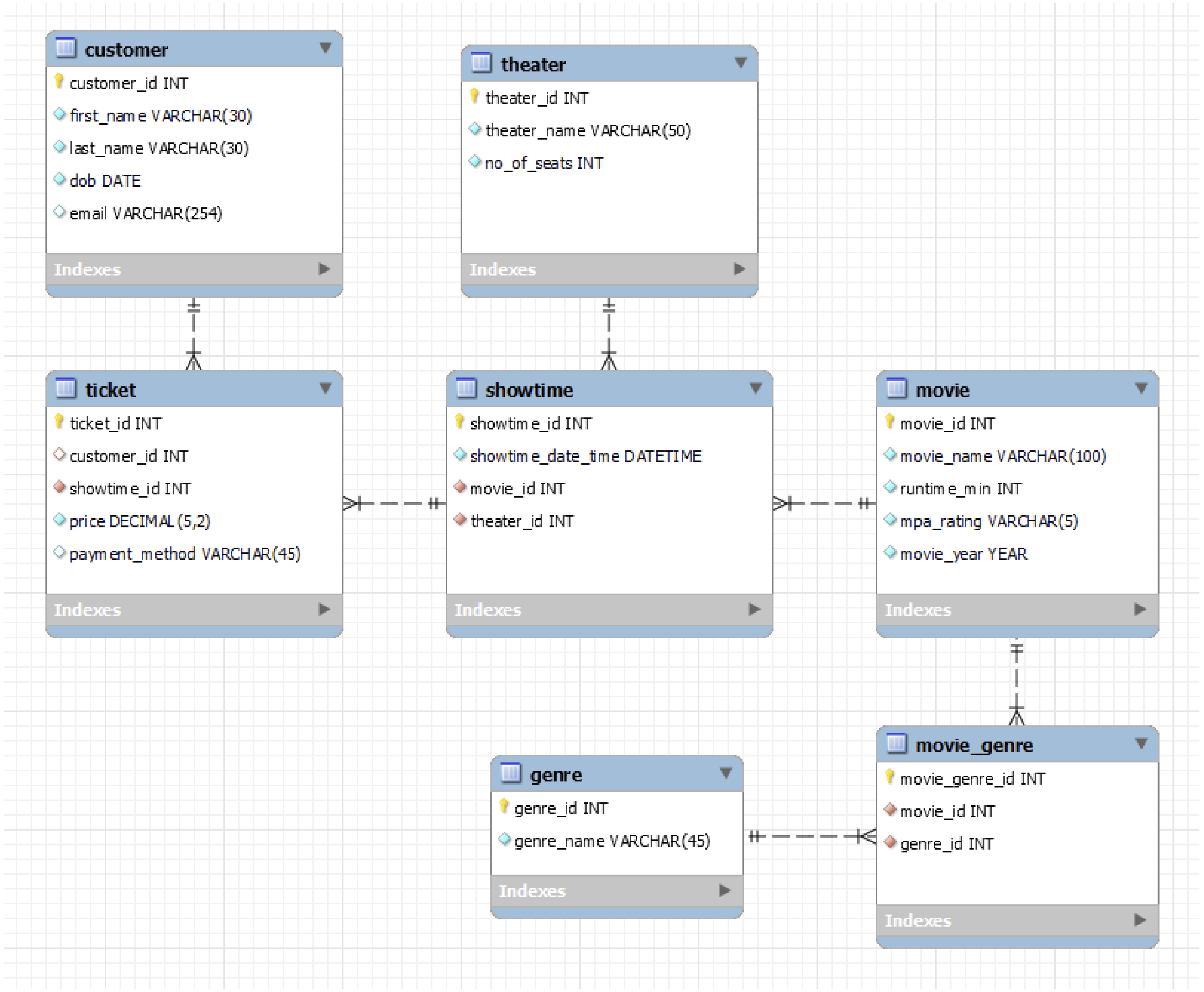
Movie_Genre		
PK	movie_genre_id NOT NULL AUTO_INCREMENT UNIQUE	int
FK	movie_id NOT NULL	int
FK	genre_id NOT NULL	int

Movie		
PK	movie_id	int NOT NULL AUTO_INCREMENT UNIQUE
Key	movie_name	varchar(100) NOT NULL
Key	runtime_min	int NOT NULL
Key	mpa_rating	varchar(5) NOT NULL
Key	movie_year	year NOT NULL

Genre		
PK	genre_id	int NOT NULL AUTO_INCREMENT UNIQUE
Key	genre_name	varchar(45) NOT NULL

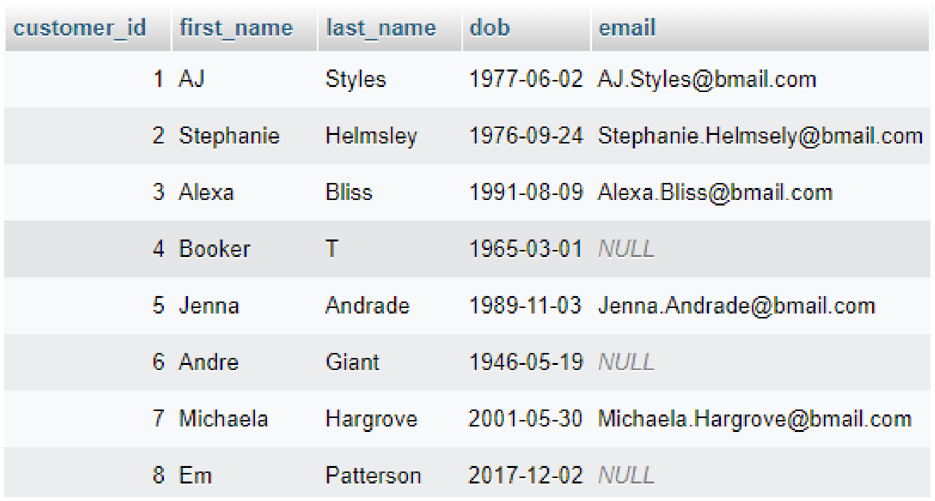
Project Implementation

Schema Diagram



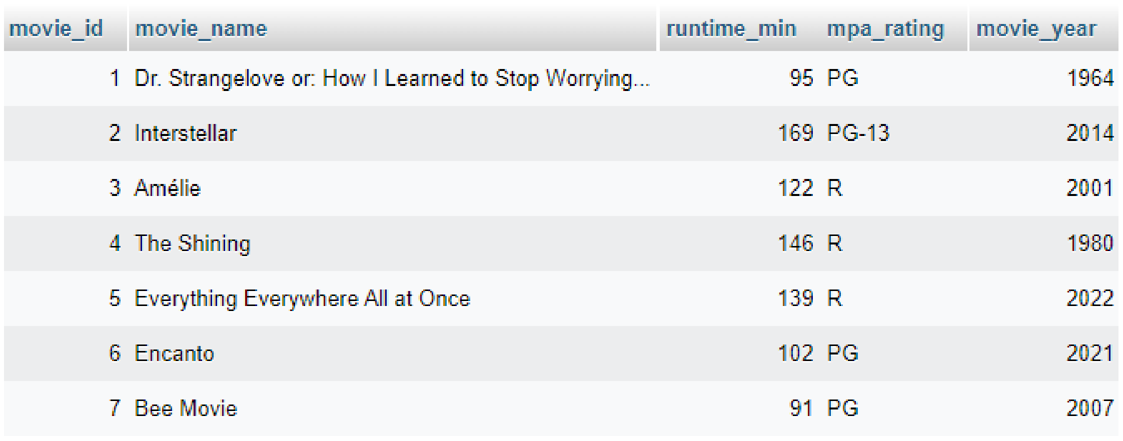
Example Data

Customer Entity's Sample Data[1]



customer_id	first_name	last_name	dob	email
1	AJ	Styles	1977-06-02	AJ.Styles@bmail.com
2	Stephanie	Helmsley	1976-09-24	Stephanie.Helmsely@bmail.com
3	Alexa	Bliss	1991-08-09	Alexa.Bliss@bmail.com
4	Booker	T	1965-03-01	*NULL*
5	Jenna	Andrade	1989-11-03	Jenna.Andrade@bmail.com
6	Andre	Giant	1946-05-19	*NULL*
7	Michaela	Hargrove	2001-05-30	Michaela.Hargrove@bmail.com
8	Em	Patterson	2017-12-02	*NULL*

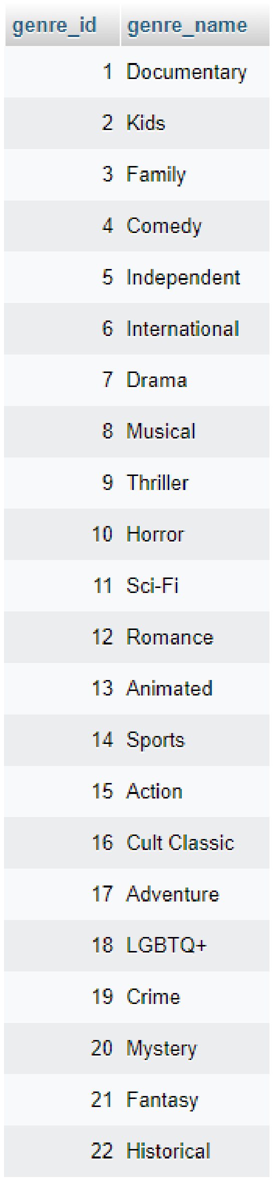
Movie Entity's Sample Data



movie_id	movie_name	runtime_min	mpa_rating	movie_year
1	Dr. Strangelove or: How I Learned to Stop Worrying...	95	PG	1964
2	Interstellar	169	PG-13	2014
3	Amélie	122	R	2001
4	The Shining	146	R	1980
5	Everything Everywhere All at Once	139	R	2022
6	Encanto	102	PG	2021
7	Bee Movie	91	PG	2007

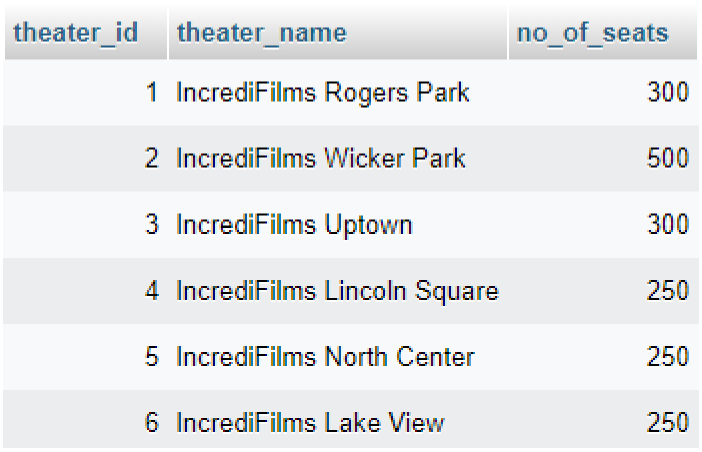
[1] Data for Customer Table is a blend of invented data and adapted data from mdabbert (2020). See Works Cited section for more details.

Genre Entity’s Sample Data



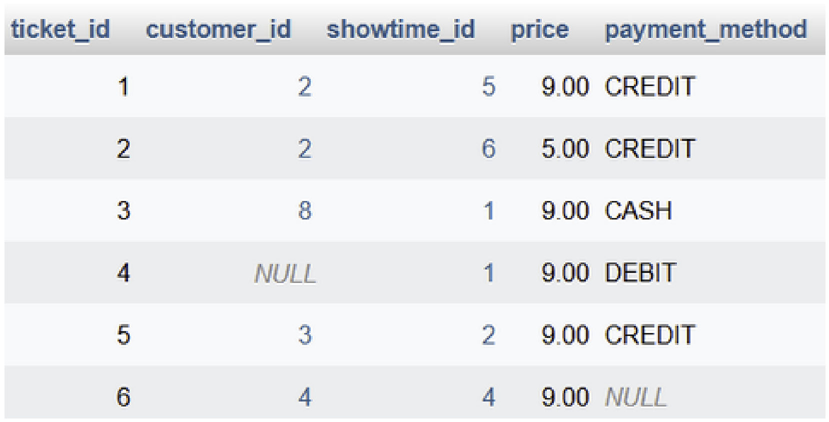
genre_id	genre_name
1	Documentary
2	Kids
3	Family
4	Comedy
5	Independent
6	International
7	Drama
8	Musical
9	Thriller
10	Horror
11	Sci-Fi
12	Romance
13	Animated
14	Sports
15	Action
16	Cult Classic
17	Adventure
18	LGBTQ+
19	Crime
20	Mystery
21	Fantasy
22	Historical

Theater Entity's Sample Data



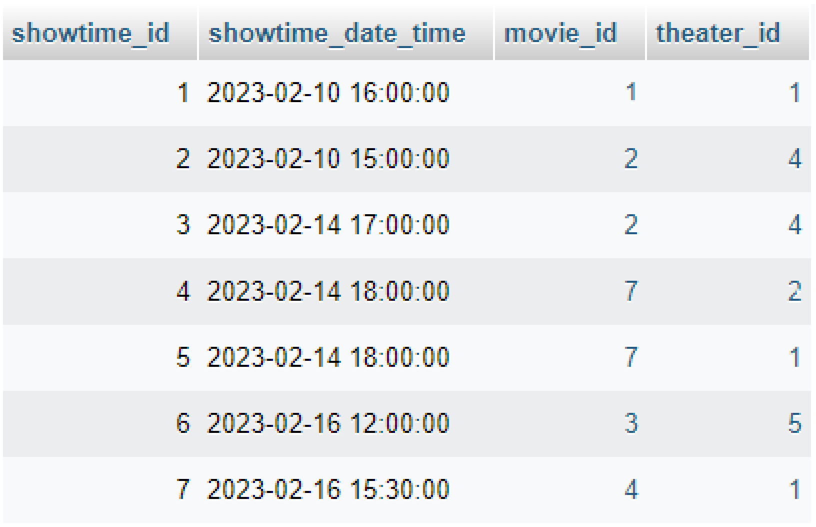
theater_id	theater_name	no_of_seats
1	IncrediFilms Rogers Park	300
2	IncrediFilms Wicker Park	500
3	IncrediFilms Uptown	300
4	IncrediFilms Lincoln Square	250
5	IncrediFilms North Center	250
6	IncrediFilms Lake View	250

Ticket Entity's Sample Data



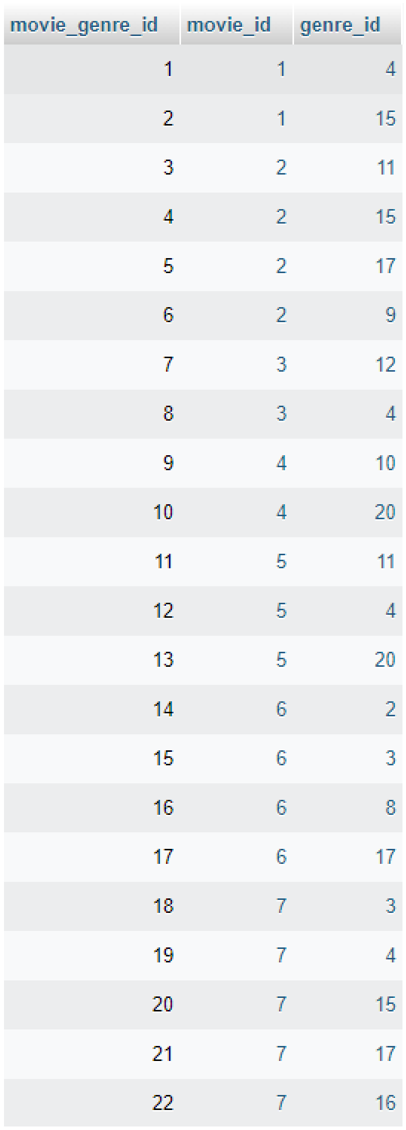
ticket_id	customer_id	showtime_id	price	payment_method
1	2	5	9.00	CREDIT
2	2	6	5.00	CREDIT
3	8	1	9.00	CASH
4	*NULL*	1	9.00	DEBIT
5	3	2	9.00	CREDIT
6	4	4	9.00	*NULL*

Showtime Entity's Sample Data



showtime_id	showtime_date_time	movie_id	theater_id
1	2023-02-10 16:00:00	1	1
2	2023-02-10 15:00:00	2	4
3	2023-02-14 17:00:00	2	4
4	2023-02-14 18:00:00	7	2
5	2023-02-14 18:00:00	7	1
6	2023-02-16 12:00:00	3	5
7	2023-02-16 15:30:00	4	1

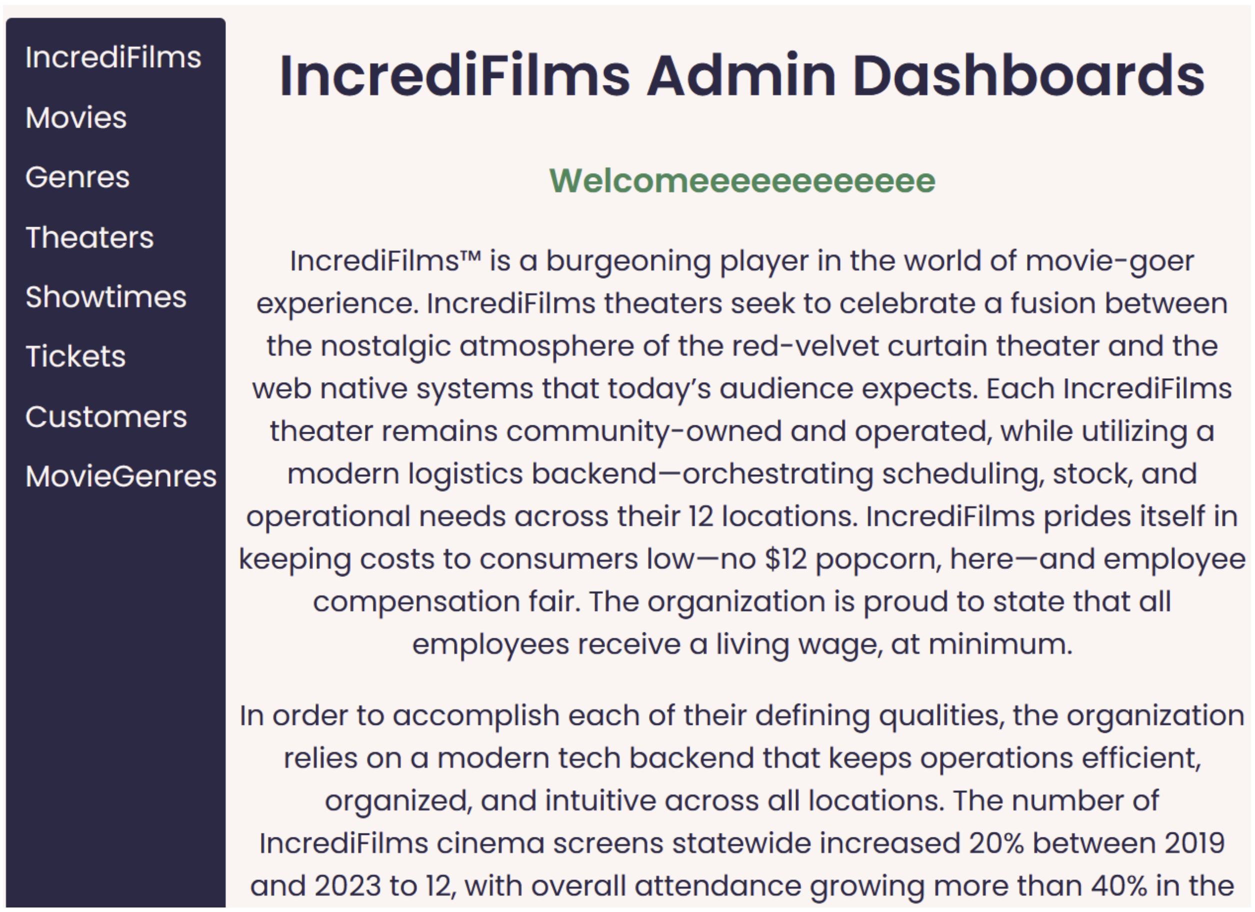
Movie_Genre’s Sample Data (Intersection Table)



movie_genre_id	movie_id	genre_id
1	1	4
2	1	15
3	2	11
4	2	15
5	2	17
6	2	9
7	3	12
8	3	4
9	4	10
10	4	20
11	5	11
12	5	4
13	5	20
14	6	2
15	6	3
16	6	8
17	6	17
18	7	3
19	7	4
20	7	15
21	7	17
22	7	16

IncrediFilms Database Manager UI

http://flip1.engr.oregonstate.edu:40593/



IncrediFilms Admin Dashboards

Welcomeeeeeeeeeeee

IncrediFilms™ is a burgeoning player in the world of movie-goer experience. IncrediFilms theaters seek to celebrate a fusion between the nostalgic atmosphere of the red-velvet curtain theater and the web native systems that today's audience expects. Each IncrediFilms theater remains community-owned and operated, while utilizing a modern logistics backend—orchestrating scheduling, stock, and operational needs across their 12 locations. IncrediFilms prides itself in keeping costs to consumers low—no $12 popcorn, here—and employee compensation fair. The organization is proud to state that all employees receive a living wage, at minimum.

In order to accomplish each of their defining qualities, the organization relies on a modern tech backend that keeps operations efficient, organized, and intuitive across all locations. The number of IncrediFilms cinema screens statewide increased 20% between 2019 and 2023 to 12, with overall attendance growing more than 40% in the

Works Cited

mdabbert. “Professional Wrestling Champions (WWE/WWF).” 2020,

https://www.kaggle.com/datasets/mdabbert/professional-wrestling-champions-wwewwf.

Retrieved on 2/5/23 via the web.