Group 29 -- Evan Maher, Jonathan Hang
Project Title: Harry Potter Database
Project Step 7 Final
Website URL: http://flip1.engr.oregonstate.edu:7719/
Please use on the Google Chrome browser

EXECUTIVE SUMMARY

Below is a summary of the major changes that were implemented throughout each step of the project.

Fixes during Week 1 of Project Development

We did not receive any feedback from peers during the first week other than praise. However, we did decide to add a Race entity to our Database Outline, because we wanted another distinct relationship. We also added relationship tables Characters_Appearances and Locations_Appearances to illustrate the relationships between Characters and Movies and between Locations and Movies.

Fixes during Week 2 of Project Development

Other than the fixes we made that were listed under each of our peers' critiques above, we also added a few more entities: Mascots, Genders, and Locations_Types. We feel that by creating these reference tables, it will be easier to modify in the future if new characters/locations are added to the series. We also fixed a relationship between Characters and Movies, and Characters and Locations where we had two relationships when the relations between the two entities are shown by the connecting entities. We also changed a few attribute names to better reflect the data that will be stored.

Fixes during Week 3 of Project Development

We received some great feedback from our peers from week 3. We added CRUD buttons to give the user UPDATE and DELETE functionality within our tables. We also added a search field under the Characters page, where the user can search by last name and the table will filter accordingly. This may be updated later on for a more advanced search that would include the character_id or another attribute. We also included preloaded "dummy data" in each table so that the user has an example to go off of when updating the database. We also received feedback from almost every peer about updating our color scheme so that the red text is easier to read against the background image. We have included a light gray background for all forms and tables, which accomplishes this task. We then added dropdown menus for house, gender, and race under the Characters page that use the FK's from other entities. These dropdowns also include a "None" value that corresponds to NULL. In addition to all of this, we have made the home page button appear in the same location on each page for consistency. In addition to everything above, we also made a variety of stylistic changes to make the site appear nicely on the screen.

Fixes during Week 4 of Project Development

Overall, we had very positive feedback from our peers this week. Not many suggestions were offered, although we did have a few that we took into consideration and made changes accordingly. We included the names of the data items in the dropdown menus and then we inserted the ID #'s into the tables below. This makes for a more user friendly approach to inserting the data. We also have many DELETE queries already working in our database, but we will have full functionality for that by the time that Step 5 is due. We hope to include functionality to delete duplicates in the tables as well. There was also an issue with our data manipulation and data definition file-s importing correctly into PHP due to the commenting style that was used to create headers. We have included a space after the first two "--" of the separation headers to fix this. An issue that we did run into was while trying to DELETE, the attributes in the parent table were set to NOT NULL which caused an SQL rejection. We then went ahead and changed that to DEFAULT NULL and where it was possible, we added ON DELETE SET NULL to the constraints as well as ON UPDATE CASCADE. We updated our database outline as well as the ERD and Schema to reflect the changes that we made.

Fixes during Week 5 of Project Development

We made some changes based on the feedback we received from our peers. We included a navigation bar that will make navigating through the subsequent pages much easier. We spent most of our time implementing both UPDATE and DELETE on almost every page, so that the project will feel more "completed". Our next goal is to implement the search and filter. In addition to the changes above, we also made it so that data was displayed in a format that abstracted foreign keys and would show the associated data instead of the FK. To keep the project files current, we updated both the DDQ, and DDM. Some struggles we ran into was implementing UPDATE on relation tables that did not contain IDs as primary. To overcome this, we had to reconsider our initial SELECT queries that utilized the alias SQL keyword and how it would affect other parts of our templates and queries. We were able to solve this issue, and everything is working as expected. Overall, we made good progress, and are quite happy with the way our project is turning out.

Fixes during Week 6 of Project Development

On the Movies page, we changed the Runtime field to only accept integer values, and the Release Date field to only accept dates (with an updated input field for date added in as well). The release date is now displayed as only a date, without the timestamp included. We also updated the tables so that they display "NULL" for null values, instead of a blank cell. The search/filter functionality is now working under the Characters page. We fixed the typo for the "Update Character" button. We removed the underscores on the home page buttons and the nav bar buttons. We changed the span backgrounds to have rounded edges for the summary text on the home page and the table titles.

PROJECT OUTLINE

Team Members: Jonathan Hang & Evan Maher

Project Title: Harry Potter Movie Database

Description: In the vast world of Harry Potter, spanning 8 separate films, there consist of over 700 characters across 4 houses. This database backend will contain a collection of data that will provide information on specific Characters, school Houses, Locations, Movie information, and Character_Movie_Appearances. Cross references between Characters and Appearances will be made, as well Characters and the Houses to which they are drafted.
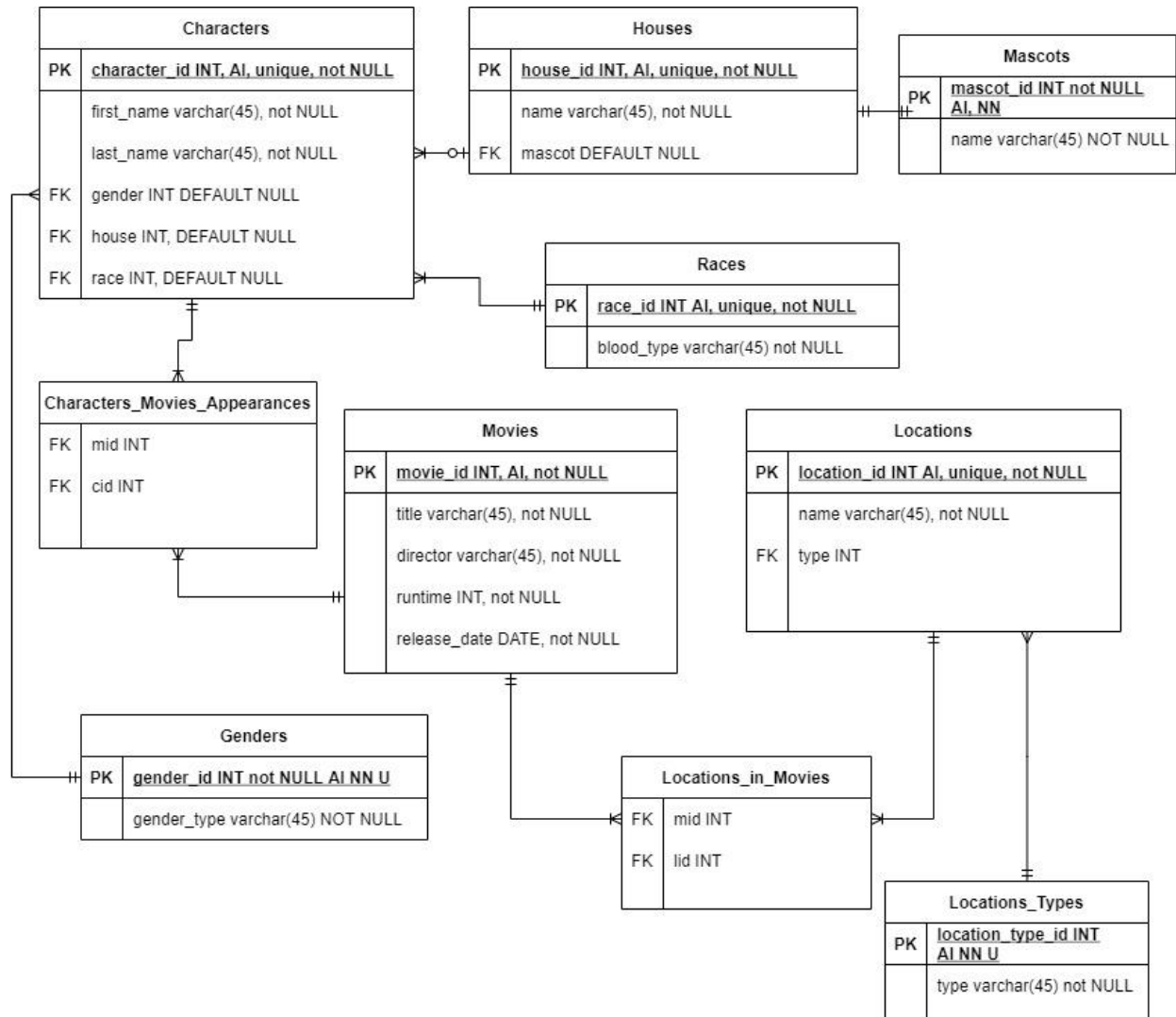
DATABASE OUTLINE

- **Characters:** records the details of Characters who appear in a movie
  - character_id: INT, auto-increment, not NULL, unique, PK
  - first_name: varchar, not NULL
  - last_name: varchar, not NULL
  - Gender: gender_id, FK DEFAULT NULL
  - house: house_id, FK DEFAULT NULL
  - race: race_id, FK DEFAULT NULL
    - Constraint: If the character does not belong to a house, then NULL.
  - Relationship:
    - M:1 relationship between Characters and Houses with house_id as a FK inside of Characters
    - M:1 relationship between Characters and Race with race_id as a FK inside of Characters
    - M:1 between Characters and Character_Movie_Appearances with character_id as a FK inside of Character_Movie_Appearances

- **Races:** records the details of characters' different blood types
  - race_id: INT, auto-increment, not NULL, unique, PK
  - blood_type: varchar, not NULL
  - Relationship:
    - M:1 relationship between Characters and Race with race_id as a FK inside of Characters

- **Houses**: records the details of the Houses that each character is sorted into

- house_id: INT, auto-increment, not NULL, unique, PK
- name: varchar, not NULL
- mascot: mascod_id, FK, DEFAULT NULL
- Relationship: 1:M relationship between Houses and Characters using house_id as the FK in Characters

- **Movies**: records the details of the Movies that each character appears in
    - movie_id: INT, auto-increment, not NULL, unique, PK
    - title: varchar, not NULL
    - director: varchar, not NULL
    - runtime (in minutes): int, not NULL
    - release_year: DATE, not NULL
    - Relationship:
        - M:1 between Movies and Character_Movie_Appearances with movie_id as a FK inside of Character_Movie_Appearances

- **Locations**: records the details of the Locations that appears in each movie
    - location_id: INT, not NULL, unique, PK
    - name: varchar, not NULL
    - type: varchar, NOT NULL
    - Relationship:
        - M:1 between Locations and Locations_in_Movies with movie_id as a FK inside of Locations_in_Movies

- **Character_Movie_Appearances**: records the details of the movies that each character appears in; this relationship table illustrates the relationship between Characters and the Movies that they appear in.
    - cid: INT, (FK)
    - mid: INT, (FK)
    - Relationship:
        - M:1 relationship between Character_Movie_Appearances and Characters
        - M:1 relationship between Character_Movie_Appearances and Movies

- **Locations_in_Movies**: records the details of the movies that each location appears in; this relationship table illustrates the relationship between Locations and the Movies that they appear in.
    - lid: int, (FK)
    - mid: INT, (FK)
    - Relationship:

- - M:1 relationship between Locations_in_Movies and Locations
  - M:1 relationship between Locations_in_Movies and Movies

- **Mascots**: records the types of mascots of each houses
  - mascot_id : int, unique, not NULL, auto-increment
  - name varchar(45) not NULL
  - Relationship: 1:1 relationship with Houses with mascot_id as a FK inside of Houses

- **Genders**: records the types of gender of the characters
  - gender_id: int, unique, not NULL, auto-increment
  - gender_type: varchar(45)
  - Relationship: 1:M with Genders and Characters and gender_id as a FK inside Characters

- **Locations_Types**: records the types of locations types
  - location_type_id: int, unique, not NULL, auto-increment
  - type: varchar(45) not NULL
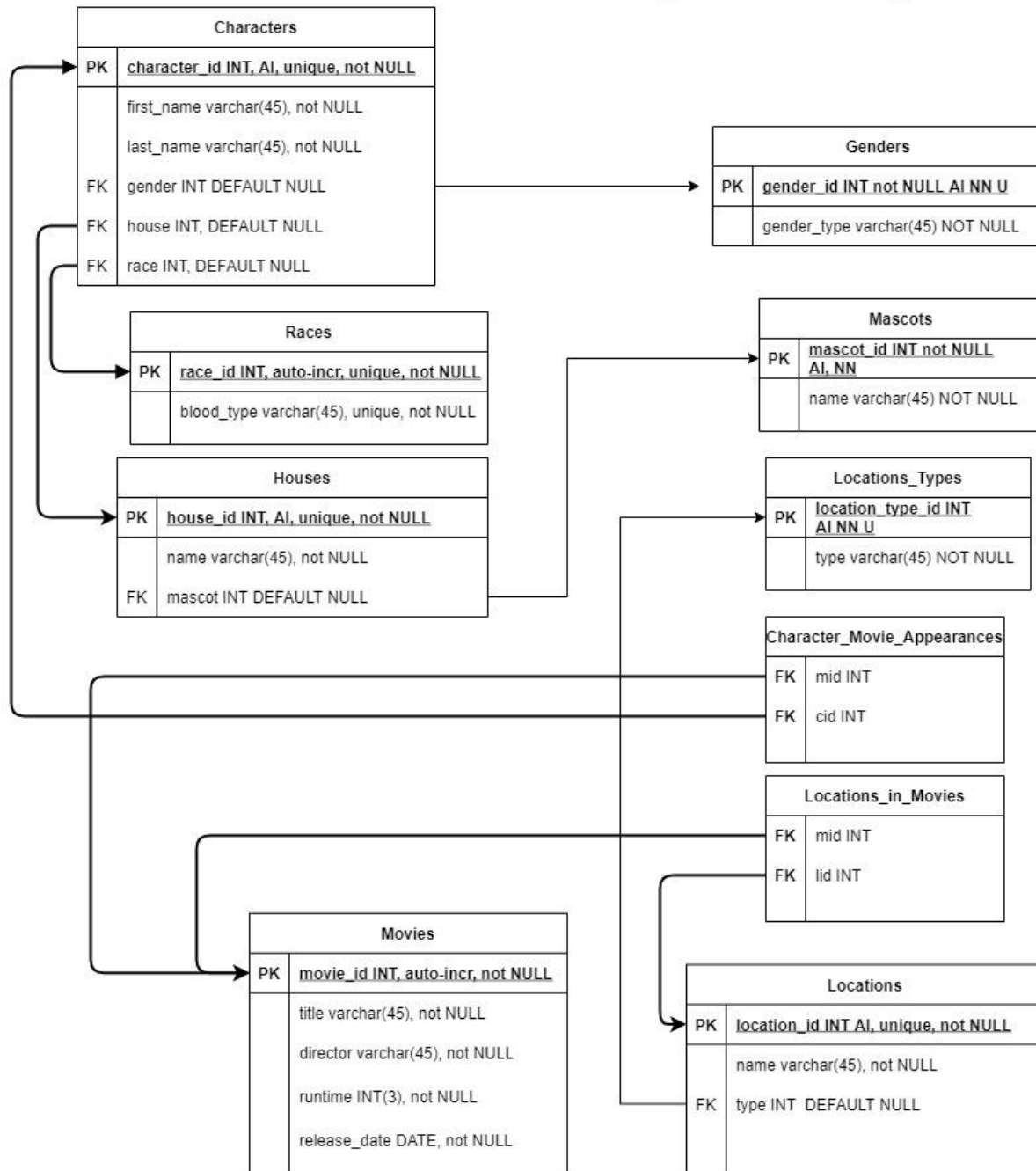 - Relationship: 1:M with Locations_Types and Locations with location_type_id as a FK inside of Locations

Entity-Relationship Diagram:

Characters

PK	character_id INT, AI, unique, not NULL
	first_name varchar(45), not NULL
	last_name varchar(45), not NULL
FK	gender INT DEFAULT NULL
FK	house INT, DEFAULT NULL
FK	race INT, DEFAULT NULL

Houses

PK	house_id INT, AI, unique, not NULL
	name varchar(45), not NULL
FK	mascot DEFAULT NULL

Mascots

PK	mascot_id INT not NULL AI, NN
	name varchar(45) NOT NULL

Races

PK	race_id INT AI, unique, not NULL
	blood_type varchar(45) not NULL

Characters_Movies_Appearances

FK	mid INT
FK	cid INT

Movies

PK	movie_id INT, AI, not NULL
	title varchar(45), not NULL
	director varchar(45), not NULL
	runtime INT, not NULL
	release_date DATE, not NULL

Locations

PK	location_id INT AI, unique, not NULL
	name varchar(45), not NULL
FK	type INT

Genders

PK	gender_id INT not NULL AI NN U
	gender_type varchar(45) NOT NULL

Locations_in_Movies

FK	mid INT
FK	lid INT

Locations_Types

PK	location_type_id INT AI NN U
	type varchar(45) not NULL

Schema:

Schema for Harry Potter Movie Database

By Evan Maher and Jonathan Hang

Characters

PK	character_id INT, AI, unique, not NULL
	first_name varchar(45), not NULL
	last_name varchar(45), not NULL
FK	gender INT DEFAULT NULL
FK	house INT, DEFAULT NULL
FK	race INT, DEFAULT NULL

Genders

PK	gender_id INT not NULL AI NN U
	gender_type varchar(45) NOT NULL

Races

PK	race_id INT, auto-incr, unique, not NULL
	blood_type varchar(45), unique, not NULL

Mascots

PK	mascot_id INT not NULL AI, NN
	name varchar(45) NOT NULL

Houses

PK	house_id INT, AI, unique, not NULL
	name varchar(45), not NULL
FK	mascot INT DEFAULT NULL

Locations_Types

PK	location_type_id INT AI NN U
	type varchar(45) NOT NULL

Character_Movie_Appearances

FK	mid INT
FK	cid INT

Locations_in_Movies

FK	mid INT
FK	lid INT

Movies

PK	movie_id INT, auto-incr, not NULL
	title varchar(45), not NULL
	director varchar(45), not NULL
	runtime INT(3), not NULL
	release_date DATE, not NULL

Locations

PK	location_id INT AI, unique, not NULL
	name varchar(45), not NULL
FK	type INT DEFAULT NULL

Home Page:

CREATE/READ/DELETE Characters Page

UPDATE Characters Page

Update Harry Potter

First Name: Harry
Last Name: Potter
Gender: Male
House: Gryffindor
Race: Half-blood

Update Character

CREATE/READ/DELETE Movies Page

Add a Movie to the Harry Potter Movies Database

*Enter movie information into fields to add a new movie *

Title: Movie's Title

Director: Director's Name

Runtime: Runtime in Minutes

Release Date: mm/dd/yyyy

Add Movie

Movies Table

Title	Director	Runtime	Release Date	
Chamber of Secrets	Chris Columbus	161 minutes	November 14 2002	Delete
Prisoner of Askaban	Alfonso Cuarón	139 minutes	June 04 2004	Delete

CREATE/READ/DELETE Houses Page

Add a House to the Harry Potter Movies Database

* Enter house information into fields to add a new house *

House Name: [House Name]

Mascot: [Lion ▾]

[Add House]

Houses Table

Name	Mascot		
Gryffindor	Lion	Update	Delete
Hufflepuff	Badger	Update	Delete
Ravenclaw	Eagle	Update	Delete
Slytherin	Snake	Update	Delete

UPDATE Houses Page

Update Ravenclaw

Name: [Ravenclaw]

Mascot: [Eagle ▾]

[Update House]

CREATE/READ/DELETE Mascots Page

Add a Mascot to the Harry Potter Movies Database

*** Enter mascot information into fields to add a new mascot ***

Name: [Name]

[Add Mascot]

Mascots Table

Name		
Lion	Update	Delete
Badger	Update	Delete
Eagle	Update	Delete
Snake	Update	Delete

UPDATE Mascots Page

Update Eagle

Mascot name: [Eagle]

[Update Mascot]

CREATE/READ/DELETE Races Page

Add a Race to the Harry Potter Movies Database

*** Enter race information into fields to add a new race ***

Race: [blood type]

[Add Race]

Races Table

Blood Type		
Muggle	Update	Delete
Half-blood	Update	Delete
Pure-blood	Update	Delete
Squib	Update	Delete
Half-breed	Update	Delete

UPDATE Races Page

Update Pure-blood

Blood Type: Pure-blood

Update Race

CREATE/READ/DELETE Genders Page

Add a Gender to the Harry Potter Movies Database

* Enter gender information into fields to add a new gender *

Gender Type: Gender Type

Add Gender

Genders Table

Gender Type		
Male	Update	Delete
Female	Update	Delete

UPDATE Genders Page

Update Male

Gender Type: Male

Update Gender

CREATE/READ/DELETE Locations Page

Add a Location to the Harry Potter Movies Database

Enter location information into fields to add a new location

Name: [Location Name]

Type: [Dwellings ▾]

[Add Location]

Locations Table

Name	Type		
Hogwarts	Schools	Update	Delete
Honeydukes	Hogsmeade	Update	Delete
Azkaban	Government-affiliated locales	Update	Delete

UPDATE Locations Page

Update Hogwarts

Location Name: [Hogwarts]

Type: [Schools ▾]

[Update Location]

CREATE/READ/DELETE Locations Types Page

Add a Location Type to the Harry Potter Movies Database

Enter location type information into fields to add a new location type

Location Type: [Location Type]

[Add Location Type]

Location Types Table

Location Type		
Dwellings	Update	Delete
Schools	Update	Delete
Diagon Alley	Update	Delete
Hogsmeade	Update	Delete
Government-affiliated locales	Update	Delete

UPDATE Locations Types Page

CREATE/READ/DELETE Locations in Movies Page

UPDATE Locations in Movies Page

Add a Character Appearance to the

Harry Potter Movies Database

** Please use on the Google Chrome browser **

Select from filters to add a new character appearance in a movie

Movie: Sorcerer's Stone ▾

Character: Harry Potter ▾

Add Character Appearance

Character Movie Appearances Table

Character ID	Movie ID	
Ron Weasley	Sorcerer's Stone	Delete
Harry Potter	Prisoner of Askaban	Delete
Rubeus Hagrid	Chamber of Secrets	Delete