Database Design

- Author: Caleb Bronn
- Last Update: 15 Mar 2025

Schema

Users

Column	Data Type	Purpose
user_id	6-digit number	Uniquely identifies the user
username	string	Username
email	string	user's email address
sign_up_date	date(YYYY-MM-DD)	Date the user signed up

SQL Table Creation

```
CREATE TABLE Users (
    user_id CHAR(6) NOT NULL PRIMARY KEY,
    username VARCHAR(30) NOT NULL,
    email VARCHAR(254) NOT NULL,    -- 254 chars is the max length
for an email
    sign_up_date DATE DEFAULT TO_DATE(CURRENT_DATE, 'YYYY-MM-DD')
);
```

Movies

Column	Data Type	Purpose
movie_id	integer	Uniquely identifies the movie based on its TMDB ID
title	string	Movie title
overview	string	Description of the movie
lang	2-digit language code	Identifies the original language of the film
bg_path	relative path to the background image	Background image of the film
poster_path	relative path to the poster image	Poster image of the film
tmdb_score	Decimal score out of 10	Average score on TMDB

Column	Data Type	Purpose
user_avg_score	Decimal score out of 10	Average score on this0 app
release_date	date(YYYY-MM-DD)	Date the film was released

SQL Table Creation

```
CREATE TABLE Movies (
    movie_id INT NOT NULL PRIMARY KEY,  -- TMDB movie IDs are
integers
    title VARCHAR(213),         -- longest movie title ever is 213
chars long
    overview VARCHAR(4000),     -- 4,000 chars is the max length for
varchar
    lang CHAR(2),               -- 2-letter language code
    bg_path VARCHAR(100),       -- relative path for movie background
image
    poster_path VARCHAR(100),   -- relative path for movie poster
image
    tmdb_score NUMBER(4, 2) CHECK(score <= 10.00),  -- scores are
stored as decimal percentages on TMDB
    user_avg_score NUMBER(4, 2) CHECK(score <= 10.00),  -- average
score from users on this app
    release_date DATE,
);
```

Reviews

Column	Data Type	Purpose
review_id	Uniquely identifies the review	
user_id	integer	Identifies the user who wrote the review
movie_id	integer	Identifies the movie the review is for
review_text	string	Review text submitted by user
your_score	Decimal score out of 10	Score submitted by user
written	date(YYYY-MM-DD)	Date the review was written
last_update	date(YYYY-MM-DD)	Date ther review was last updated

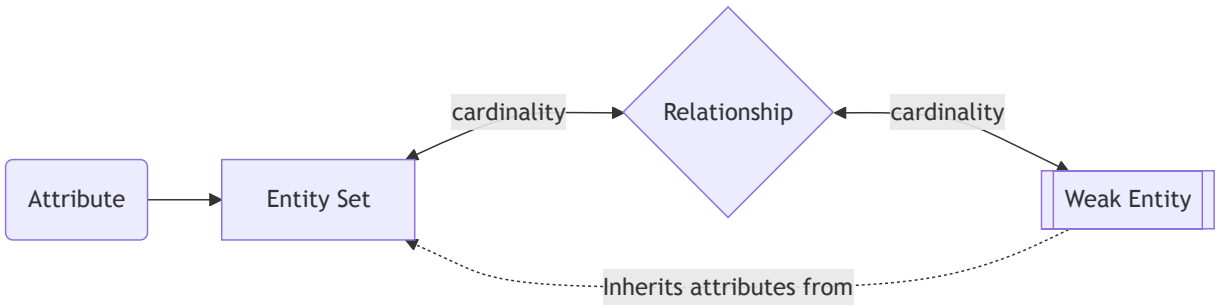
SQL Table Creation

```
CREATE TABLE Reviews (
    review_id CHAR(9) NOT NULL PRIMARY KEY,
    user_id CHAR(6) FOREIGN KEY REFERENCES Users(user_id),
    movie_id INT FOREIGN KEY REFERENCES Movies(movie_id),
    review_text VARCHAR(4000) NOT NULL, --4,000 chars is the max
length for varchar
    your_score NUMBER(4, 2) CHECK(score <= 10.00),
    written DATE DEFAULT TO_DATE(CURRENT_DATE, 'YYYY-MM-DD'),
    last_update DATE                    -- same as 'written' if
review has never been updated
);
```

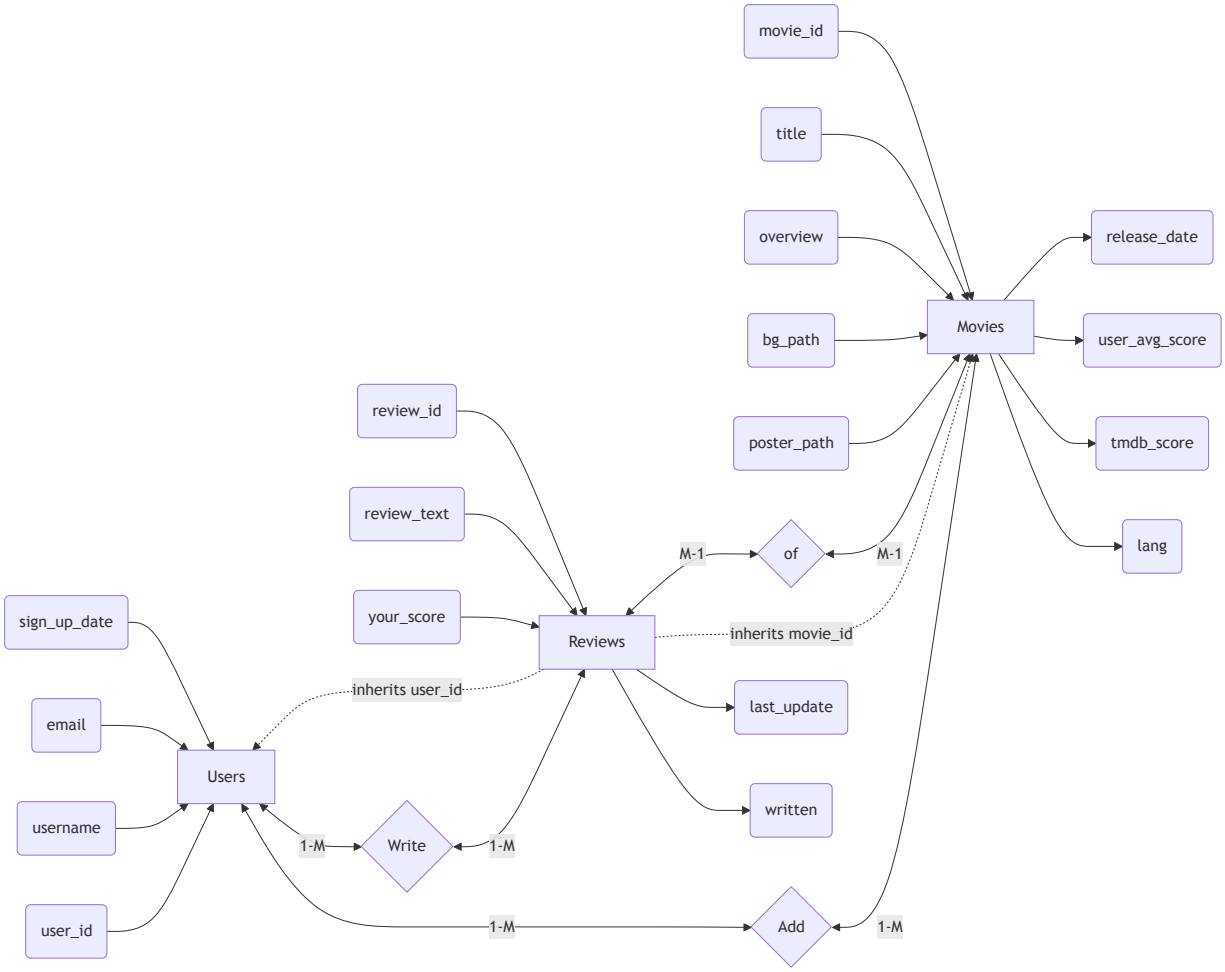
Entity-Relationship Diagram

This diagram displays the attributes and relationships between each table in the database.

Legend



ERD


movie_id
title
overview
release_date
Movies
bg_path
user_avg_score
review_id
poster_path
tmdb_score
review_text
lang
M-1
of
M-1
sign_up_date
your_score
Reviews
inherits movie_id
inherits user_id
email
last_update
Users
written
username
1-M
Write
1-M
user_id
1-M
Add
1-M