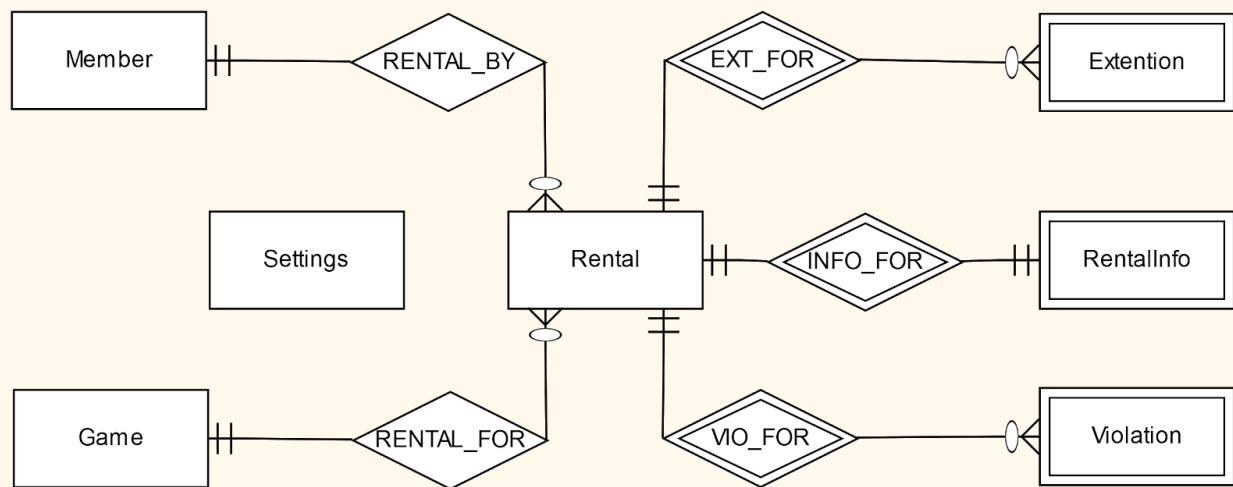


Document Design



The created database videogames_db attempts to construct a rental system that is easy to manage as the catalogue of members grows. Members are identified by unique id's (they also have names and emails for correspondence purposes). Members also have active balances and a status that defines whether they are banned or not.

Games have basic information that is displayed through tables on the site and are also identified by a unique id.

The rental system centers around a Rental entry which, although it can be safely deleted, doesn't ever have to be. Rentals are a combination of a memberid and a gameid, identified by its own unique id, rentalid. Whether a rental is active or not separates an 'archive' of rentals from ongoing rentals. Through this it is easy to construct 'logs' where staff can access the previous rental histories of members.

RentalInfo is dependant on a Rental and is deleted upon a rental becoming inactive. This is where the information important to the rental is held, such as the start and due date.

Extention, like rentalInfo, keeps track of extentions per rental. And violations does the same for breaking the society's rules. This way it is easy to see the state of a rental in its entirety,

including violations, dates, extentions, and other information. Clearing all the information associated with a single rental is handled neatly with a cascade delete through all of the above tables.

The settings table is only accessible when the database is accessed by the secretary, or someone with root privileges. This keeps track of all the secretaries settings iterations, and thus you can either reference the most recent iteration to pull the current restrictions and rules, or revert to previous more desirable versions that will all be kept within the database. This also helps new secretaries get a feel for the restrictions previous secretaries used. We have added some games to the database to demonstrate the system.

File Structure

Our current file structure is akin to this.

- Main
 - Private
 - Shared
 - Dashboard.php
 - Header.php
 - Footer.php
 - Credentials.php
 - Database.php
 - Functions.php
 - Initialize.php
 - Sql_functions.php
 - Public
 - Staff
 - Games
 - Create.php
 - Delete.php
 - Details.php
 - Edit.php
 - Records.php
 - New.php
 - Members

- Delete.php
 - Edit.php
 - Records.php
 - New.php
 - Violation.php
- Rentals
 - Ban.php
 - Extend.php
 - New.php
 - Records.php
- Stylesheets
 - KobeGames.css
 - Bg.png
- Index.php
- Access.sql
- Videogames_db.sql

From the file structure, we placed respective files into Private and Public folders for efficiency as well as code clarity. The private folder includes scripts at set up the database, initialize it, as well as run queries on it. We also have shared folder which contains a dashboard, a header and a footer since they're shared headlessly across the entire website.

As for our public folder, we have an index.php, a staff folder and our stylesheets. The staff folder contains a respective folder for each of the task the staff member is supposed to carry :- manage rentals, manage the video game database and finally, manage the members of the community. We decided to use separate php files for all the functions necessary to carry them out for efficiency's sake.

Contributors

- Wei Chew
- Alessandro Amantini
- Joshua O'Connor
- Sanchit Bembi
- Yilei Liang
- Shuhel Ahmed