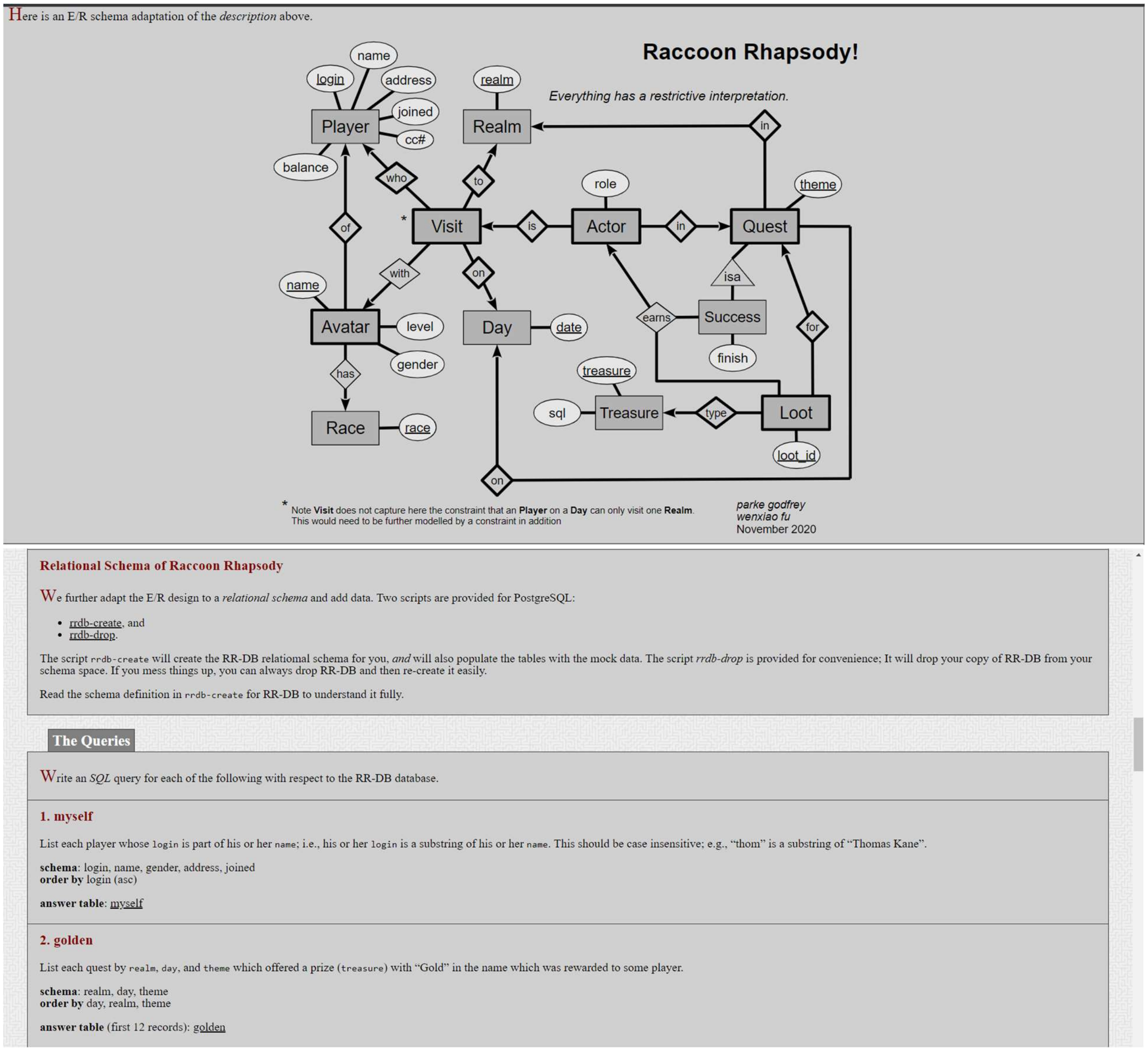


Here is an E/R schema adaptation of the *description* above.

* Note **Visit** does not capture here the constraint that an **Player** on a **Day** can only visit one **Realm**. This would need to be further modelled by a constraint in addition

parke godfrey
wenxiao fu
November 2020

Relational Schema of Raccoon Rhapsody

We further adapt the E/R design to a *relational schema* and add data. Two scripts are provided for PostgreSQL:

- rrdb-create, and
- rrdb-drop.

The script `rrdb-create` will create the RR-DB relatiomal schema for you, *and* will also populate the tables with the mock data. The script *rrdb-drop* is provided for convenience; It will drop your copy of RR-DB from your schema space. If you mess things up, you can always drop RR-DB and then re-create it easily.

Read the schema definition in `rrdb-create` for RR-DB to understand it fully.

The Queries

Write an *SQL* query for each of the following with respect to the RR-DB database.

1. myself

List each player whose `login` is part of his or her `name`; i.e., his or her `login` is a substring of his or her `name`. This should be case insensitive; e.g., “thom” is a substring of “Thomas Kane”.

schema: login, name, gender, address, joined
order by login (asc)

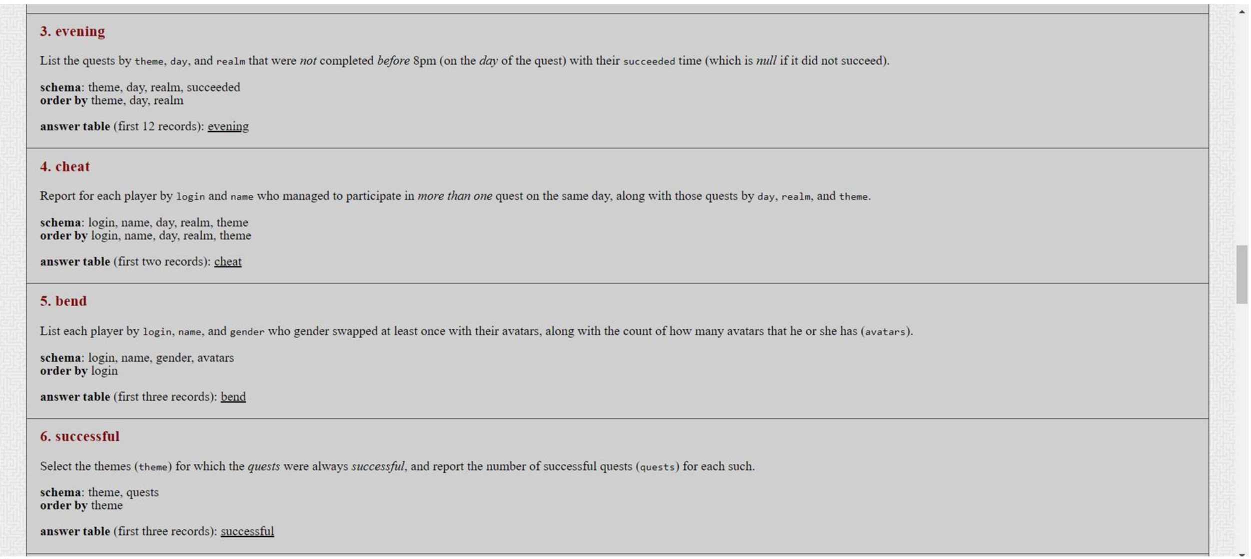
answer table: myself

2. golden

List each quest by `realm`, `day`, and `theme` which offered a prize (`treasure`) with “Gold” in the name which was rewarded to some player.

schema: realm, day, theme
order by day, realm, theme

answer table (first 12 records): golden



3. evening

List the quests by `theme`, `day`, and `realm` that were *not* completed *before* 8pm (on the *day* of the quest) with their `succeeded` time (which is *null* if it did not succeed).

schema: theme, day, realm, succeeded
order by theme, day, realm

answer table (first 12 records): evening

4. cheat

Report for each player by `login` and `name` who managed to participate in *more than one* quest on the same day, along with those quests by `day`, `realm`, and `theme`.

schema: login, name, day, realm, theme
order by login, name, day, realm, theme

answer table (first two records): cheat

5. bend

List each player by `login`, `name`, and `gender` who gender swapped at least once with their avatars, along with the count of how many avatars that he or she has (`avatars`).

schema: login, name, gender, avatars
order by login

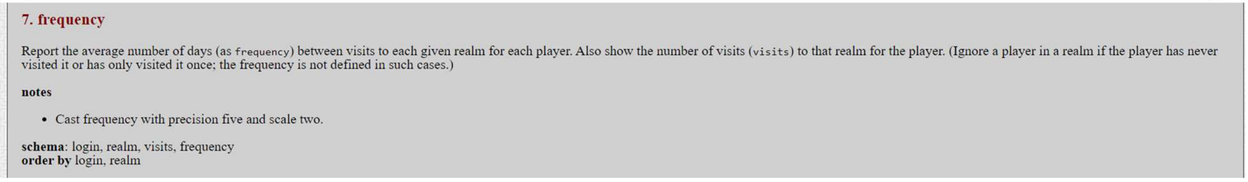
answer table (first three records): bend

6. successful

Select the themes (`theme`) for which the *quests* were always *successful*, and report the number of successful quests (`quests`) for each such.

schema: theme, quests
order by theme

answer table (first three records): successful



7. frequency

Report the average number of days (as `frequency`) between visits to each given realm for each player. Also show the number of visits (`visits`) to that realm for the player. (Ignore a player in a realm if the player has never visited it or has only visited it once; the frequency is not defined in such cases.)

notes

- Cast frequency with precision five and scale two.

schema: login, realm, visits, frequency
order by login, realm