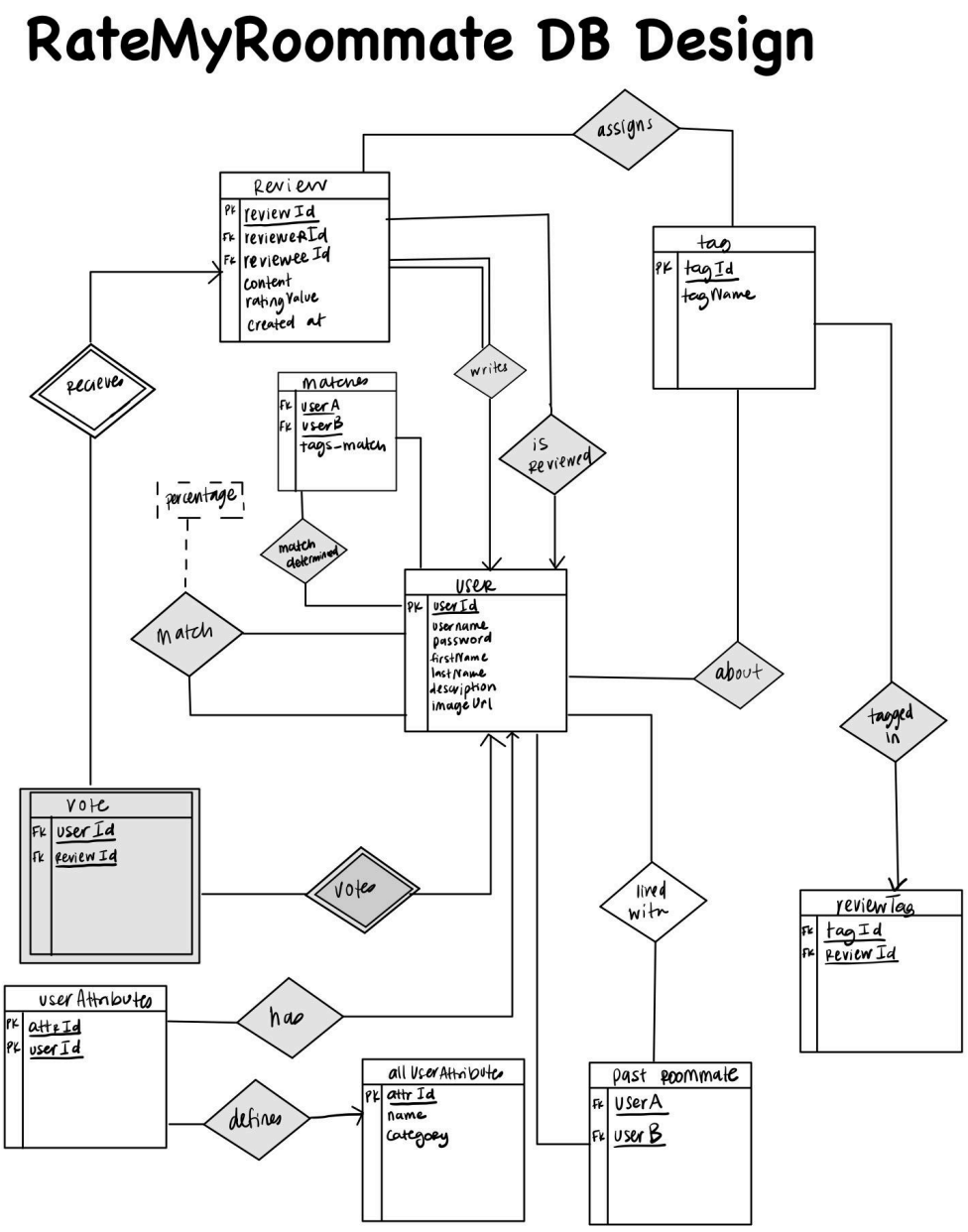
ER Model


RateMyRoommate DB Design
assigns
Review
PK reviewId
FK reviewerId
FK revieweeId
content
ratingValue
created at
tag
PK tagId
tagName
receives
writes
matches
FK userA
FK userB
tags-match
is reviewed
percentage
match determined
User
PK userId
username
password
firstName
lastName
description
imageUrl
match
about
tagged in
Vote
FK userId
FK reviewId
votes
lived with
reviewTag
FK tagId
FK reviewId
userAttributes
PK attrId
PK userId
has
defines
allUserAttributes
PK attrId
name
category
pastRoommate
FK userA
FK userB

Resulting Relations

```
user(userId, username, password, firstName, lastName, description, imageUrl)
pk: (userId)
not null: username, password, firstName, lastName
unique: (username)

review(reviewId, reviewerId, revieweeId, content, ratingValue, createdAt)
pk: (reviewId)
not null: reviewerId, revieweeId, content, ratingValue
fk: (reviewerId) referencing user(userId)
fk: (revieweeId) referencing user(userId)

vote(userId, reviewId, type)
pk: (userId, reviewId)
not null: userId, reviewId
fk: (userId) referencing user(userId)
fk: (reviewId) referencing review(reviewId)

tag(tagId, tagName)
pk: (tagId)
not null: tagName

reviewTag(tagId, reviewId)
pk: (tagId, reviewId)
not null: tagId, reviewId
fk: (tagId) referencing tag(tagId)
fk: (reviewId) referencing review(reviewId)

allUserAttributes(attrId, name, category)
pk: (attrId)
not null: name
unique: (name)

userAttributes(attrId, userId)
pk: (userId, attrId)
not null: attrId, userId
fk: (userId) referencing user(userId)
fk: (attrId) referencing allUserAttributes(attrId)

matches(userA, userB, tags_match)
pk: (userA, userB)
not null: userA, userB
fk: (userA) referencing user(userId)
fk: (userB) referencing user(userId)

pastRoommate(userA, userB)
pk: (userA, userB)
not null: userA, userB
fk: (userA) referencing user(userId)
fk: (userB) referencing user(userId)
```

Functional Dependencies

user(userId, username, password, firstName, lastName, description, imageUrl)

- userId → username, password, firstName, lastName, description, imageUrl
- username → userId, password, firstName, lastName, description, imageUrl

review(reviewId, reviewerId, revieweeId, content, ratingValue, createdAt)

- reviewId → reviewerId, revieweeId, content, ratingValue, createdAt

vote(userId, reviewId, type)

- (userId, reviewId) → type

tag(tagId, tagName)

- tagId → tagName

reviewTag(tagId, reviewId)

- (tagId, reviewId) → (no non-key attributes; only trivial dependencies)

allUserAttributes(attrId, name, category)

- attrId → name, category
- name → attrId, category

userAttributes(attrId, userId)

- (userId, attrId) → (no non-key attributes; only trivial dependencies)

matches(userA, userB, tags_match)

- (userA, userB) → tags_match

pastRoommate(userA, userB)

- (userA, userB) → (no non-key attributes; only trivial dependencies)

Normalization

user(userId, username, password, firstName, lastName, description, imageUrl)
Functional Dependencies:

- userId → username, password, firstName, lastName, description, imageUrl
- username → userId, password, firstName, lastName, description, imageUrl

Both userId and username are candidate keys.
→ Table is in BCNF.

review(reviewId, reviewerId, revieweeId, content, ratingValue, createdAt)
Functional Dependency:

- reviewId → reviewerId, revieweeId, content, ratingValue, createdAt
  reviewId is the primary key.

Table is in BCNF.

vote(userId, reviewId, type)
Functional Dependency:

- (userId, reviewId) → type
  (userId, reviewId) is the primary key.

Table is in BCNF.

tag(tagId, tagName)
Functional Dependency:

- tagId → tagName
  tagId is the primary key.

Table is in BCNF.

reviewTag(tagId, reviewId)
No non-trivial FDs (composite key, no additional attributes).
Table is in BCNF.

allUserAttributes(attrId, name, category)
Functional Dependencies:

- attrId → name, category
- name → attrId, category
  Both attrId and name are candidate keys.

Table is in BCNF.

userAttributes(attrId, userId)
No non-trivial FDs (composite key, no additional attributes).
Table is in BCNF.

matches(userA, userB, tags_match)
Functional Dependency:

- (userA, userB) → tags_match
  (userA, userB) is the primary key.

Table is in BCNF.

pastRoommate(userA, userB)
No non-trivial FDs (composite key, no additional attributes).
Table is in BCNF.