

3NF DECOMP

attributes: {userid, username, password, fname, lname, isbn, title, author, category, description, year, reviewId, body, liked, created_date, follows, wishlisted}

Functional Dependencies:

userid -> username, password, fname, lname
isbn -> title, author, category, description, year
reviewId -> body, liked, created_date, userId, isbn, author
userId, isbn -> reviewId, like, body, created_date
authorId -> author
userId, isbn -> wishlisted

Step 1:

userId -> username
userId -> password
userId -> fname
userId -> lname

isbn -> title
isbn -> author
isbn -> category
isbn -> description
isbn -> year

reviewId -> body
reviewId -> liked
reviewId -> created_date
reviewId -> userId
reviewId -> isbn
reviewId -> author

userId, isbn -> reviewId
userId, isbn -> like
userId, isbn -> body
userId, isbn -> created_date
userId, isbn -> wishlisted

authorId -> author

Step 2:

userId -> username

userId -> password

userId -> fname

userId -> lname

isbn -> title

isbn -> author

isbn -> category

isbn -> description

isbn -> year

reviewId -> body

reviewId -> liked

reviewId -> created_date

reviewId -> userId

reviewId -> isbn

reviewId -> author

userId, isbn -> reviewId

userId+ = {userId, username, password, fname, lname}

X

isbn+ = {isbn, title, author, category, description, year}

X

userId, isbn -> like

X

userId, isbn -> body

X

userId, isbn -> created_date

X

userId, isbn -> wishlisted

X

authorId -> author

Step 3:

userId -> username

userId -> password

userId -> fname

userId -> lname

isbn -> title

isbn -> author

isbn -> category

isbn -> description
isbn -> year

reviewId -> body
reviewId -> liked
reviewId -> created_date
reviewId -> userId
reviewId -> isbn
reviewId -> author

userId, isbn -> reviewId
userId, isbn -> like {userId, isbn}+ = {userId, isbn, username, password, fname, lname, title, author, category, description, year, reviewid, like} X
 userId, isbn -> reviewId implies that any userId, isbn can determine the same RHS as reviewId.

userId, isbn -> body X
userId, isbn -> created_date X
userId, isbn -> wishlisted

authorId -> author

Minimal Cover:

userId -> username
userId -> password
userId -> fname
userId -> lname
isbn -> title
isbn -> author
isbn -> category
isbn -> description
isbn -> year
reviewId -> body
reviewId -> liked
reviewId -> created_date
reviewId -> userId
reviewId -> isbn
reviewId -> author
userId, isbn -> reviewId
userId, isbn -> wishlisted

authorId -> author

merge LHS:

userId -> username, password, fname, lname

isbn -> title, author, category, description, year

reviewId -> body, liked, created_date, userId, isbn

userId, isbn -> reviewId, wishlisted

authorId -> author

Create Tables:

user(userId, username, password, fname, lname)

book(isbn, title, authorId, category, description, year)

review(reviewId, body, liked, created_date, userId, isbn)

wishlist(userId, isbn, wishlisted)

author(authorId, author)

Candidate key {reviewId} NOT IN ANY TABLE

Final Tables:

user(userId, username, password, fname, lname)

book(isbn, title, authorId, category, description, year)

review(reviewId, body, liked, created_date, userId, isbn)

wishlist(userId, isbn, wishlisted)

author(authorId, author)

key(reviewId)

Caveats:

books can have multiple authors

Not every user is going to have a review