Genre(Name, Overview)

- Name -> Overview
- Based on the right hand rule all possible ck is (**name**)
    - Name -> Name, Overview
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**overview**) fully functionally dependent on a candidate key (**name**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key

Book(ISBN, Title, Publish_Date, Description)

- ISBN -> Title, Publish_Date, Description
- Based on the right hand rule all possible ck is (**ISBN**)
    - ISBN -> Title, Publish_Date, Description
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**Title, Publish_Date, Description**) fully functionally dependent on a candidate key (**ISBN**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key

Author(Author_ID, FName, LName, Bio)

- Author_ID -> FName, LName, Bio
- Based on the right hand rule all possible ck is (**Author_ID**)
    - Author_ID -> FName, LName, Bio
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**FName, LName, Bio**) fully functionally dependent on a candidate key (**Author_ID**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key

User(User_Id, Fname, Lname, DOB, Status, password, email)

- User_ID -> Fname, Lname, DOB, Status, password, email
- Based on the right hand rule all possible ck is (**User_ID**)
    - User_ID -> Fname, Lname, DOB, Status, password, email
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**Fname, Lname, DOB, Status, password, email**) fully functionally dependent on a candidate key (**User_ID**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key

UserRole(roleID, Role_name)

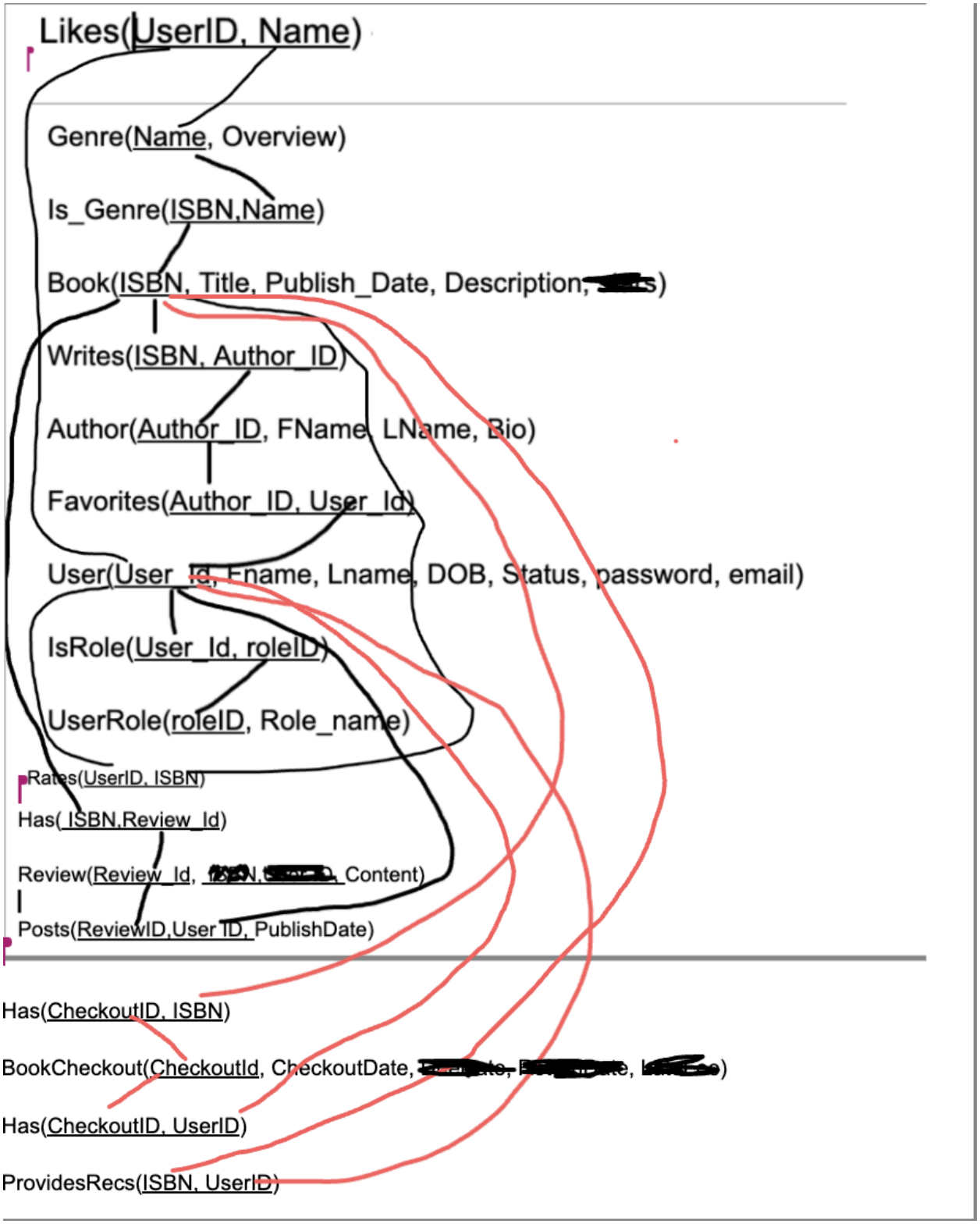
- RoleID -> Role_name
- Based on the right hand rule all possible ck is (**roleID**)
    - roleID -> Role_name
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**Role_name**) fully functionally dependent on a candidate key (**roleID**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key

Review(Review_Id,Content)

- Review_ID -> Content
- Based on the right hand rule all possible ck is (**review_ID**)
    - review_ID -> Content
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**Content**) fully functionally dependent on a candidate key (**review_ID**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key

BookCheckout(CheckoutId, CheckoutDate)

- CheckoutID -> CheckoutDate
- Based on the right hand rule all possible ck is (**CheckoutID**)
    - CheckoutID -> CheckoutDate
- 1st normal as all attributes are defined as atomic
- 2nd normal as all non candidate keys (**CheckoutDate**) fully functionally dependent on a candidate key (**CheckoutID**)
- 3rd normal as no non candidate key attribute is transitively dependent on a candidate key
- BCNF as every functional dependency contains a candidate key


Likes(UserID, Name)
Genre(Name, Overview)
Is_Genre(ISBN,Name)
Book(ISBN, Title, Publish_Date, Description, [redacted])
Writes(ISBN, Author_ID)
Author(Author_ID, FName, LName, Bio)
Favorites(Author_ID, User_Id)
User(User_Id, Fname, Lname, DOB, Status, password, email)
IsRole(User_Id, roleID)
UserRole(roleID, Role_name)
Rates(UserID, ISBN)
Has(ISBN,Review_Id)
Review(Review_Id, [redacted], Content)
Posts(ReviewID,User ID, PublishDate)
Has(CheckoutID, ISBN)
BookCheckout(CheckoutId, CheckoutDate, [redacted])
Has(CheckoutID, UserID)
ProvidesRecs(ISBN, UserID)

Genre(Name, Overview)
- Name -> Overview

Is_Genre(ISBN,Name)

Book(ISBN, Title, Publish_Date, Description)

- ISBN -> Title, Publish_Date, Description

Writes(ISBN, Author_ID)

Author(Author_ID, FName, LName, Bio)
- Author_ID -> FName, LName, Bio

Favorites(Author_ID, User_Id)

User(User_Id, Fname, Lname, DOB, Status, password, email)
- User_ID -> Fname, Lname, DOB, Status, password, email

IsRole(User_Id, roleID)

UserRole(roleID, Role_name)
- RoleID -> Role_name

Rates(UserID, ISBN)

Has(ISBN,Review_Id)

Review(Review_Id,Content)
- Review_ID -> Content

Posts(ReviewID,User ID, PublishDate)

Has(CheckoutID, ISBN)

BookCheckout(CheckoutId, CheckoutDate)
- CheckoutID -> CheckoutDate

Has(CheckoutID, UserID)

ProvidesRecs(ISBN, UserID)