User

(username, password, join\_date, bio, profile\_picture)

Candidate key: username, password

FDs: username, password → join\_date, bio, profile\_picture

Follows

(follower, followee)

Candidate key: follower, followee

FDs: N/A

Post(ID, name, publish\_date, short\_description, steps, difficulty, time, author)

Candidate key: ID

FDs: ID → name, publish\_date, short\_description, steps, difficulty, time, author

Photo(postID, url)

Candidate key: postID, url

FDs: N/A

Rate(username, postID, value)

Candidate key: username, postID

FDs: username, postID → value

User\_Likes(username, postID)

Candidate key: username, postID

FDs: N/A

Bookmark(username, postID)

Candidate key: username, postID

FDs: N/A

Comment(author, postID, textual\_content, date)

Candidate key: username, date

FDs: username, date → postID, textual\_content

Tag(tag\_name)

Candidate key: tag\_name

FDs: N/A

Has\_Tag(postID, tag\_name)

Candidate key: postID, tag\_name

FDs: N/A

Ingredient(ingredient\_name)

Candidate key: ingredient\_name

FDs: N/A

Contains\_Ingredient(postID, ingredient\_name, value, unit)

Candidate key: postID, ingredient\_name

FDs: postID, ingredient\_name → value, unit

All the relations are in BCNF because they contain all fully functional dependencies with the candidate key determining all non-candidate key attributes, all transitive dependencies are decomposed, and for all functional dependencies, the candidate key is on the left side. In addition, functional dependencies were preserved when we decomposed into BCNF.