Vinit Kirit Jain

1017922770

2023-09-05

Social Media Database Project Proposal

Introduction:

In today's interconnected world, social media has become the heartbeat of our digital lives. It's where individuals, communities, and businesses converge to share, connect, and communicate. At the core of social media experience is the user. The user can create an account and make posts that include text and pictures. They can follow other users to stay updated and can also exchange messages privately.

Every time a user logs in, their login details are stored in the Login table with their timestamps, device type and IP addresses. When creating a new account, the user is asked to enter their unique username and password, as well as a bio section where they can enter some content about themselves. Their specific timestamps and a unique numeric user id will be generated at the time of profile creation. Users on this platform are allowed to make posts sharing their life updates with the help of photos and text. A post comprises of many attributes like post id that gets uniquely generated to identify a singular post. Each post also contains the caption/text, creation time and the location of where it was posted. Users can react to each post which can be a like, or dislike or any other emoji. They can add a comment to the post to share their thoughts where we will store the comment text, creation time and whether the comment was edited. It is a Boolean field that holds true value if the comment has been edited.

Entities:

Users:

Attributes: user_id (primary key), user_name, email, password, account_bio, creation_time

Users represent individuals who have registered on the social media platform. User profiles typically include information such as usernames, display names, bios, and contact details. Users create and interact with posts, follow other users, and engage in various social activities on the platform.

Posts:

Attributes: post_id (primary key), location, creation_time, caption.

Content published by individuals on social media platforms is known as posts. They may contain timestamps, text, pictures, or videos. Users can share updates, ideas, and multimedia content through posts with their followers and the larger community.

Comments:

Attributes: comment_id, comment_text, was_edited, creation_time.

User-generated responses to posts are known as comments. Users can leave comments on posts to share their ideas, viewpoints, or responses. Text, timestamps, and user information are frequently included in comments.

Reactions:

Attributes: reaction_time, reaction_type

User opinions on postings and comments are represented via reactions. With the use of likes, thumbs up/down, and other indicators, users can respond to content. Users can quickly express their views or ideas about a post or comment by reacting to it.

Photos:

Attributes: photo_id (primary key), photo_url, creation_time, size

Users can submit photos to their profiles or include them into postings as multimedia elements. These pictures are added to posts.

Login:

Attributes: device_type, login_time, ip_address

Users can login to their account multiple times and each login info is stored in this table along with the timestamp, device type and ip address.

Relationships:

Creates relationship: Users creating posts.

Contains relationship: Posts contains comments.

Have relationship: Posts have reactions.

Consist_of relationship: Posts consists of photos.

Send relationship: Users sends messages to other users.

Follow relationship: Users follow other users.

Example Transactions:

Insert a new user account.

Count all the comments on a post.

Select a list of liked posts and order it by number of likes.

Select a list of posts and order by creation time.

Select a list of all the chats between two users.

Insert a new photo inside a post.

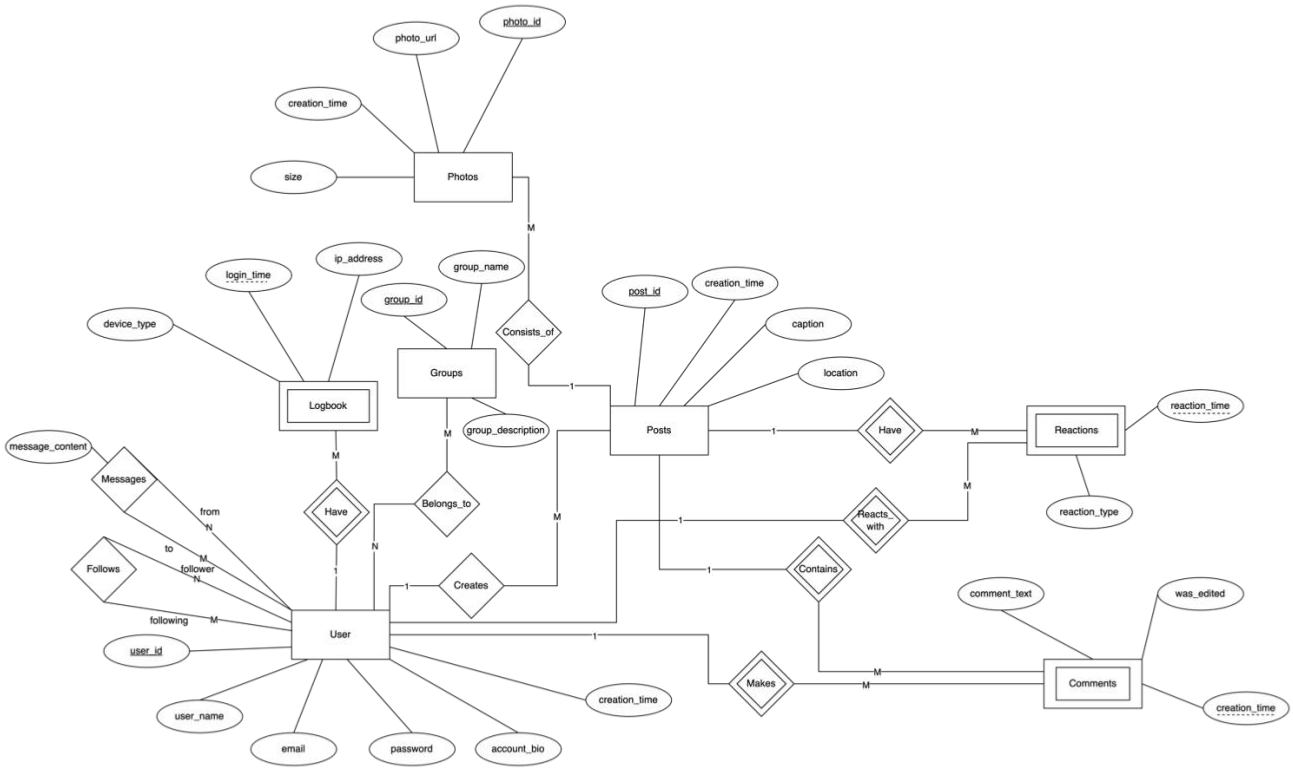
Changing the username/password.

Order all comments on a post by number of reactions.

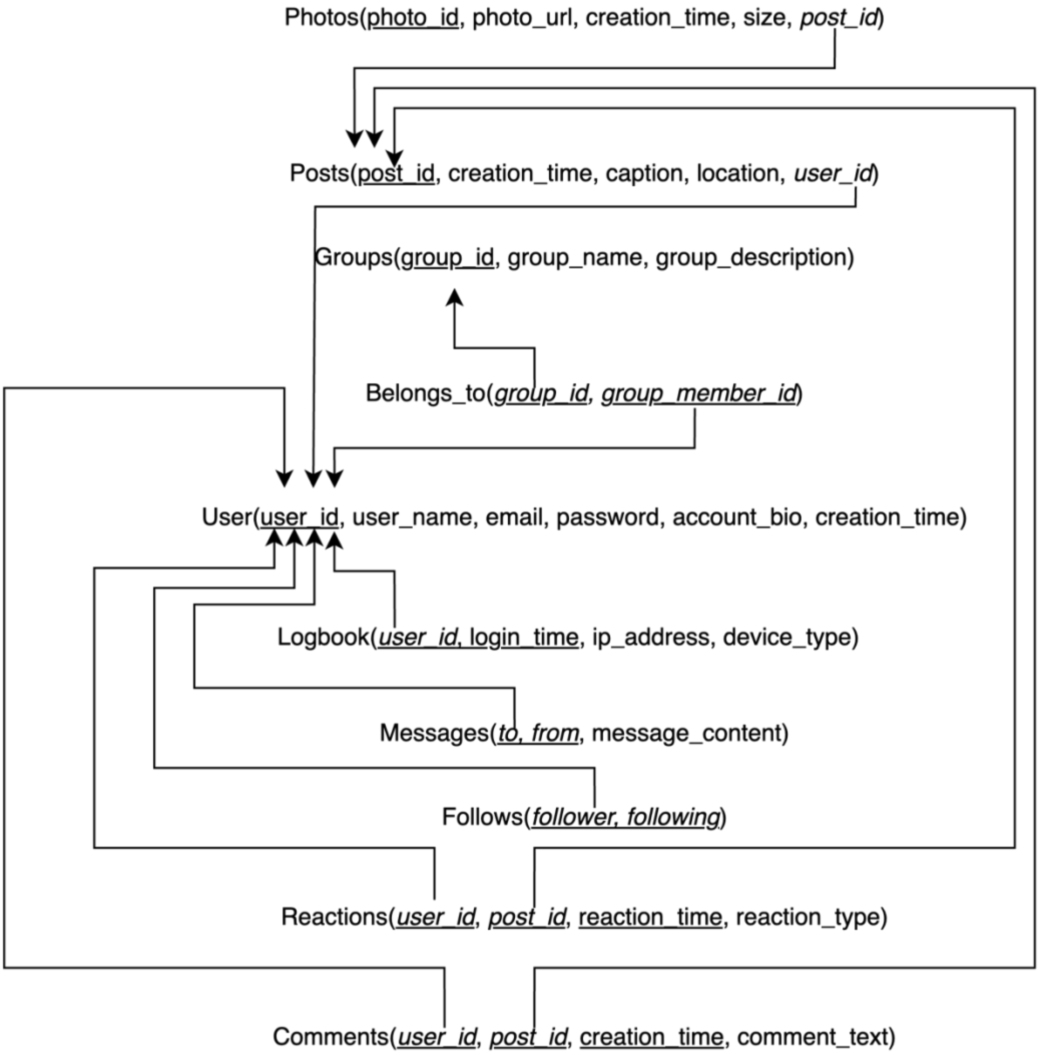
Count all the messages between two users.

Update username and password

ER diagram: -



Relational Schema: -



Photos(photo_id, photo_url, creation_time, size, *post_id*)

Posts(post_id, creation_time, caption, location, *user_id*)

Groups(group_id, group_name, group_description)

Belongs_to(*group_id, group_member_id*)

User(user_id, user_name, email, password, account_bio, creation_time)

Logbook(*user_id*, login_time, ip_address, device_type)

Messages(*to, from*, message_content)

Follows(*follower, following*)

Reactions(*user_id*, *post_id*, reaction_time, reaction_type)

Comments(*user_id*, *post_id*, creation_time, comment_text)

Normalization: -

Table: - Photos (photo_id, photo_url, creation_time, size, *post_id*)

Candidate Key: {photo_id}

Primary Key: {photo_id}

Functional Dependencies:

{photo_id} -> {photo_url, creation_time, size, post_id}

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

Here the key only has one attribute, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key"

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case photo_id is a super key, so the table is in BCNF.

Table: - Posts (post_id, creation_time, caption, location, user_id)

Candidate Key: { post_id }

Primary Key: { post_id }

Functional Dependencies:

{ post_id } -> {creation_time, caption, location, user_id}

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

4. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
5. No non-key attribute is FD on just part of the key.
6. If key has only one attribute, and R is 1NF, R is automatically 2NF.

Here the key only has one attribute, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

3. To be 3NF, relation must be 2NF and have no transitive dependencies.
4. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

2. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
3. Stricter than 3NF, which allows A to be part of a candidate key.
4. If there is just one single candidate key, the forms are equivalent.

In this case post_id is a super key, so the table is in BCNF.

Table: - Groups (group_id, group_name, group_description)

Candidate Key: { group_id }

Primary Key: { group_id }

Functional Dependencies:

{ group_id } -> { group_name, group_description }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

Here the key only has one attribute, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes “candidate key”.

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case group_id is a super key, so the table is in BCNF.

Table: - Groups (group_id, group_name, group_description)

Candidate Key: { group_id }

Primary Key: { group_id }

Functional Dependencies:

{ group_id } -> { group_name, group_description }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

Here the key only has one attribute, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case group_id is a super key, so the table is in BCNF.

Table: - Belongs_to(group_id, group_member_id)

Candidate Key: { group_id, group_member_id }

Primary Key: { group_id, group_member_id }

Functional Dependencies:

{ group_id, group_member_id } -> { group_id, group_member_id }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

All the non-key attributes are fully functionally dependent on the key, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case group_id, group_member_id are super keys, so the table is in BCNF.

Table: - User (user_id, user_name, email, password, account_bio, creation_time)

Candidate Key: { user_id }

Primary Key: { user_id }

Functional Dependencies:

{ user_id } -> { user_name, email, password, account_bio, creation_time }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

Here the key only has one attribute, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
2. If there is just one single candidate key, the forms are equivalent.

In this case user_id are super keys, so the table is in BCNF.

Table: - Messages (to, from, message_content)

Candidate Key: { to, from }

Primary Key: { to, from }

Functional Dependencies:

{ to, from } -> { message_content }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

All the non-key attributes are fully functionally dependent on the key, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case to, from are super keys, so the table is in BCNF.

Table: - Follows (follower, following)

Candidate Key: { follower, following }

Primary Key: { follower, following }

Functional Dependencies:

{ follower, following } -> { follower, following }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

All the non-key attributes are fully functionally dependent on the key, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case follower, following are super keys, so the table is in BCNF.

Table: - Reactions (user_id, post_id, reaction_time, reaction_type)

Candidate Key: { user_id, post_id }

Primary Key: { user_id, post_id }

Functional Dependencies:

{ user_id, post_id } -> { user_id, post_id, reaction_time, reaction_type }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

All the non-key attributes are fully functionally dependent on the key, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case user_id, post_id are super keys, so the table is in BCNF.

Table: - Comments (user_id, post_id, creation_time, comment_text)

Candidate Key: { user_id, post_id }

Primary Key: { user_id, post_id }

Functional Dependencies:

{ user_id, post_id } -> { user_id, post_id, creation_time, comment_text }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

All the non-key attributes are fully functionally dependent on the key, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case user_id, post_id are super keys, so the table is in BCNF.

Table: - Logbook (user_id, login_time, ip_address, device_type)

Candidate Key: { user_id, login_time }

Primary Key: { user_id, login_time }

Functional Dependencies:

{ user_id, login_time } -> { , ip_address, device_type }

1st Normal Form:

Rules:

1. A relation is in 1NF if and only if every attribute is single valued for each tuple.

All the attributes in the table are single valued so the table is already in the 1st Normal Form.

2nd Normal Form:

Rules:

1. A relation is in second normal form (2NF) if it is in first normal form and all the non-key attributes are fully functionally dependent on the key.
2. No non-key attribute is FD on just part of the key.
3. If key has only one attribute, and R is 1NF, R is automatically 2NF.

All the non-key attributes are fully functionally dependent on the key, so the table is in the 2nd normal form.

3rd Normal Form:

Rules:

1. To be 3NF, relation must be 2NF and have no transitive dependencies.
2. No non-key attribute determines another non-key attribute. Here key includes "candidate key".

There are no transitive dependencies, so the table is in the 3rd normal form.

Boyce-Codd Normal Form:

Rules:

1. A relation is in Boyce/Codd Normal Form (BCNF) if whenever a non-trivial functional dependency X→A exists, then X is a super key.
2. Stricter than 3NF, which allows A to be part of a candidate key.
3. If there is just one single candidate key, the forms are equivalent.

In this case user_id, post_id are super keys, so the table is in BCNF.