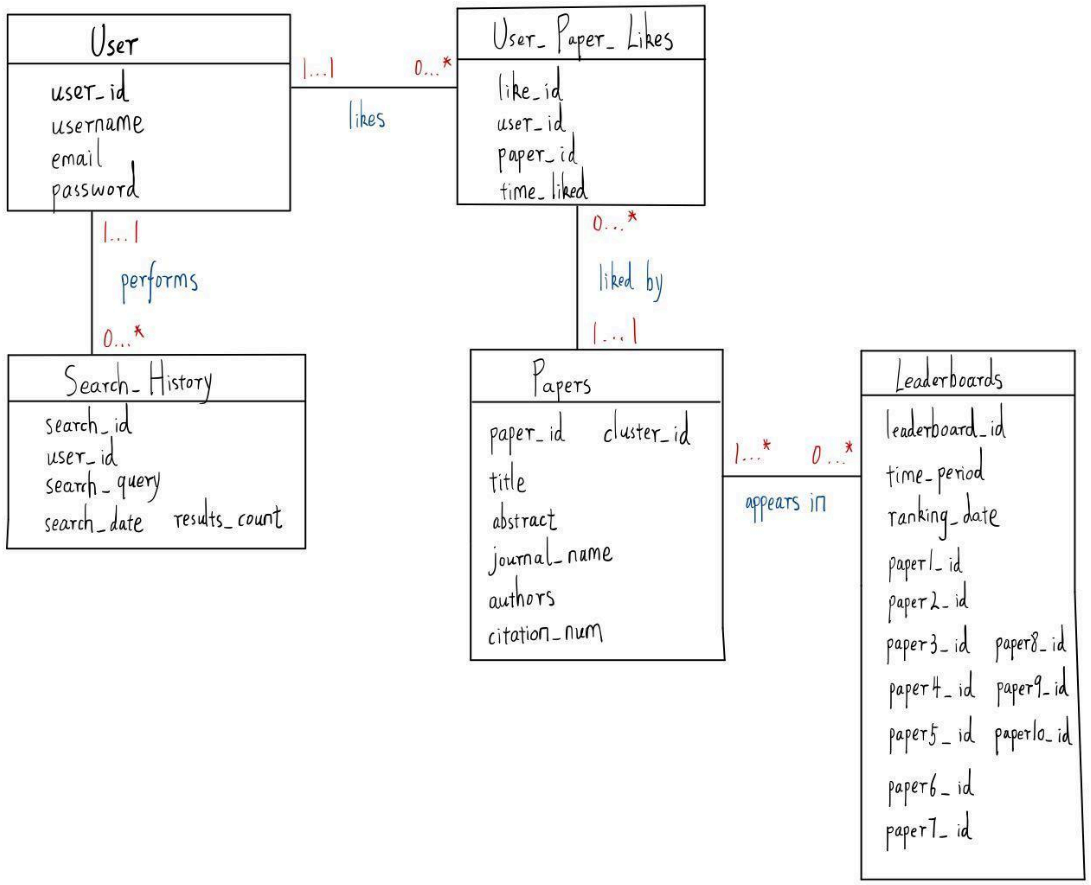
PART 2:

Let’s walk through the creation of a UML diagram for your "FindMyPaper" project, following the assumptions and relationships based on the information provided.

1.



2. Assumptions and Entity Descriptions

We modeled the ‘User’ entity as an entity because it represents the information of the people interacting with the platform. It contains attributes such as ‘username’, ‘email’, and ‘password’. Each user can perform multiple actions like searches and likes, which is why we have one-to-many relationships with User_Paper_Likes and Search_History.

The ‘Search History’ entity tracks the user’s search queries; it is stored as an entity because it helps both us and the users to better track the searches they performed. While a user can perform zero or many searches, each search will only belong to one user. Therefore, we have a one-to-many relationship between ‘User’ and ‘Search_History’.

The ‘User_Paper_Likes’ entity keeps track of which users have liked which papers and when. It is an entity because while there is a many-to-many relationship between users and papers, we still need to store the additional information such as ‘time_liked’, which would not be manageable as an attribute of ‘User’ or ‘Papers’. For cardinality, each action is unique to a combination of user and paper. Thus, ‘User_Paper_Likes’ has one-to-many relationships with ‘User’ and ‘Papers’.

The ‘Papers’ entity represents the information of each paper, containing attributes such as ‘title’, ‘abstract’, ‘journal_name’, ‘authors’, and ‘citation_count’. It is an important entity because the attributes are essential for the basic functions of our program. A paper can be liked 0 to many times and can appear in 0 to multiple leaderboards.

Lastly, the ‘Leaderboards’ entity contains information about each leaderboard, as well as the date and time the paper was ranked and the period for ranking, i.e. one week or one month. It also includes the IDs of the top ten papers. It is constructed as an entity because it represents the ranking feature of our program. It has a many-to-many relationship with ‘Papers’ because a paper can appear in multiple leaderboards and each leaderboard can feature multiple papers over different periods.

3. See the table above

4. See above because this is already in BCNF

1. User Table

Primary Key: user_id

Potential Candidate Keys: Assuming username and email are unique, they can also serve as candidate keys.

Functional Dependencies:
user_id → username, email, password
username → user_id, email, password
email → user_id, username, password

Analysis:
All determinants (user_id, username, email) are superkeys.

BCNF Compliance: Satisfied

2. Search_History Table
Primary Key: search_id

Foreign Key: user_id references User

Functional Dependencies:
search_id → user_id, search_query, search_date, results_count

Analysis:
search_id is the sole determinant and a superkey.

BCNF Compliance: Satisfied

3. Papers Table
Primary Key: paper_id

Potential Issues: Attributes like cluster_id, journal_name, or authors might have dependencies.

Functional Dependencies:
paper_id → cluster_id, title, abstract, journal_name, authors, citation_num

Analysis:
Assuming no non-trivial dependencies exist where non-superkeys determine other attributes, the table is in BCNF.
Since cluster_id does not determine other attributes this does not violate BCNF.

BCNF Compliance: Satisfied

4. . User_Paper_Likes Table

Primary Key: like_id

Foreign Keys: user_id references User, paper_id references Papers

Functional Dependencies:
like_id → user_id, paper_id, time_liked

Analysis:
like_id is the sole determinant and a superkey.

BCNF Compliance: Satisfied

5 . Leaderboards Table

Primary Key: leaderboard_id

Functional Dependencies:
leaderboard_id → paper1_id, paper2_id, ..., other attributes

Analysis:
leaderboard_id functionally determines all other attributes in the table.
No other functional dependencies exist among non-key attributes.

Superkey Verification:
Since leaderboard_id is the primary key, it inherently qualifies as a superkey.

Non-Superkey Determinants:
There are no non-trivial functional dependencies where a non-superkey attribute determines another attribute. For instance, attributes like paper1_id, paper2_id, etc., do not determine any other attributes within the table.

BCNF Compliance:
Every determinant in the table is a superkey, so the table meets all BCNF criteria.

5. logical design (relational schema)

- User(user_id: INT [PK], username: VARCHAR(X), email: VARCHAR(X), password: VARCHAR(X))

- Papers(paper_id: INT [PK], title: VARCHAR(X), abstract: TEXT, journal_name: VARCHAR(X), authors: VARCHAR(X), citation_num: INT)

- User_Paper_Likes(like_id: INT [PK], user_id: INT [FK to User.user_id], paper_id: INT [FK to Papers.paper_id], time_liked: DATETIME)

- Search_History(search_id: INT [PK], user_id: INT [FK to User.user_id], search_query: VARCHAR(X), search_date: DATETIME, results_count: INT)

- Leaderboards(leaderboard_id: INT [PK], time_period: INT, ranking_date: DATETIME, paper1_id: INT[FK to Papers.paper_id], paper2_id: INT[FK to Papers.paper_id], paper3_id: INT[FK to Papers.paper_id], paper4_id: INT[FK to Papers.paper_id], paper5_id: INT[FK to Papers.paper_id], paper6_id: INT[FK to Papers.paper_id], paper7_id: INT[FK to Papers.paper_id], paper8_id: INT[FK to Papers.paper_id], paper9_id: INT[FK to Papers.paper_id], paper10_id: INT[FK to Papers.paper_id])