DATABASE Team Project

Tutor: Nagat Drawel

Team member：Wei Zhu, Lei Fu, Jianwen Wu, Hua Zhang

Journal of IoT Research (JIR)

Database Modeling

October 8, 2022

VANIER
CÉGEP/COLLEGE
Continuing Education

CONTENTS

Introduction

The Journal of IoT Research (JIR) is a prestigious information systems research journal. It uses a peer-review process to select research papers for publication. Only about 20% of the papers submitted to the journal are accepted for publication. A new issue of the journal is published each quarter. The editor would like you and your team to design a database system to manage the publication process. The requirements data to be modelled are as follows:

1. Research papers are submitted by scholars. When a paper is received, the editor will assign a unique number, and record some basic information about it in the system. The title of the paper, the date it was received, and a paper status of "received" are entered.

2. The editor will briefly review the topic of the paper to ensure that its contents fall within the scope of the journal. If the content is not within the scope of the journal, the paper's status is changed to "rejected" and the author (scholar) is notified via e-mail. If the content is within the scope of the journal, then the editor selects three or more reviewers to review the paper.

3. Information about the scholar(s) is also recorded. For each scholar, the name, mailing address, e-mail address, and affiliation (school or company for which the author works) is recorded.

4. Every paper must have an author. Only authors that have submitted paper are kept in the system. It is typical for one paper to have several authors (scholars). A single author may have submitted many different papers to the journal. Moreover, when a paper has multiple authors, it is important to record the order in which the authors are listed in the paper credits.

5. Reviewers work for other companies or universities and read papers to ensure the scientific validity of the paper. In addition to basic reviewers' information, the system records their areas of interest. Areas of interest are pre-defined areas of expertise that the reviewer has specified. An area of interest is identified by an IS code and includes a description (e.g. IS2003 is the code for "database modeling").

6. A reviewer can have many areas of interest, and an area of interest can be associated with many reviewers. All reviewers must specify at least one area of interest. It is unusual, but it is possible to have an area of interest for which the journal has no reviewers.

7. A reviewer will typically receive several papers to review each year, although new reviewers may not have received any paper yet.

8. The editor will change the status of the paper to “under review” and record which reviewers the paper was sent to and the date on which it was sent to each reviewer. The process started when the reviewers read the paper and provide feedback to the editor regarding the paper.

9. The feedback from each reviewer includes rating the paper on a 10-point scale for appropriateness, clarity, methodology, and contribution to the field, as well as a recommendation for publication (accept or reject).

10. The editor will record all this information in the system for each review received from each reviewer and the date that the feedback was received. Once all of the reviewers have provided their evaluation of the paper, the editor will decide whether or not to publish the paper. If the editor decides to publish the paper, the paper’s status is changed to “accepted” and the date of acceptance for the paper is recorded. If the paper is not to be published, the status is changed to “rejected.”

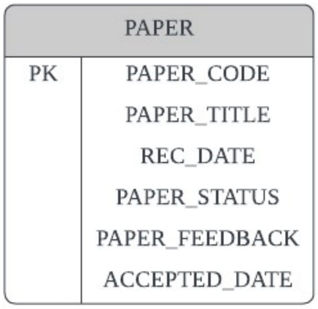
According to the given information, our team designed the Entity Relationship (E/R) diagram, shows entity types, relationships, attributes, key attributes, cardinalities, and participation constraints. Also, normalize the entities and include the steps of normalization process (to 3NF), After that we create database schema using SQL DDL statements, show how they enforced the constraints implied by their logical design, identify any constraints that are not captured by the ER diagram, and we also apply advanced database concepts such as triggers to maintain data integrity, indexes, views, and procedures to improve system performance.

Conceptual Design of the Database

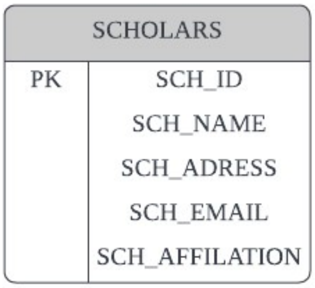
For creating the ER diagram, we have the following steps:

Step 1: Entities and Attributes

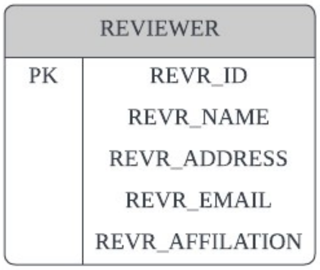
Based on given information, we identify 5 entities and their initial attributes.



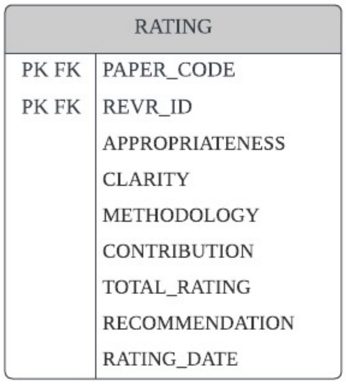
PAPER	
PK	PAPER_CODE
	PAPER_TITLE
	REC_DATE
	PAPER_STATUS
	PAPER_FEEDBACK
	ACCEPTED_DATE



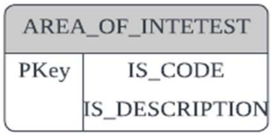
SCHOLARS	
PK	SCH_ID
	SCH_NAME
	SCH_ADRESS
	SCH_EMAIL
	SCH_AFFILATION



REVIEWER	
PK	REVR_ID
	REVR_NAME
	REVR_ADDRESS
	REVR_EMAIL
	REVR_AFFILATION



RATING	
PK FK	PAPER_CODE
PK FK	REVR_ID
	APPROPRIATENESS
	CLARITY
	METHODOLOGY
	CONTRIBUTION
	TOTAL_RATING
	RECOMMENDATION
	RATING_DATE

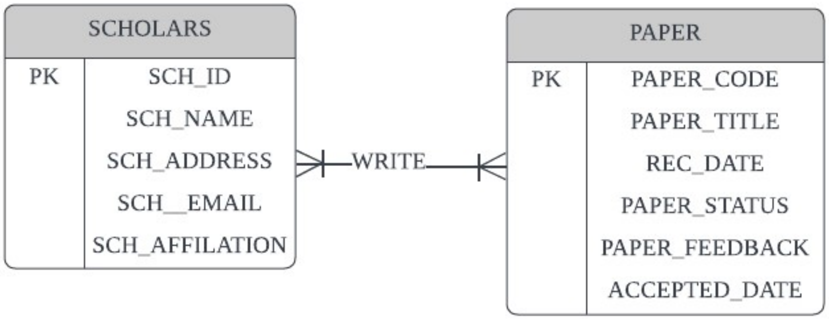


AREA_OF_INTETEST	
PKey	IS_CODE
	IS_DESCRIPTION

Step 2: Relationship Analysis

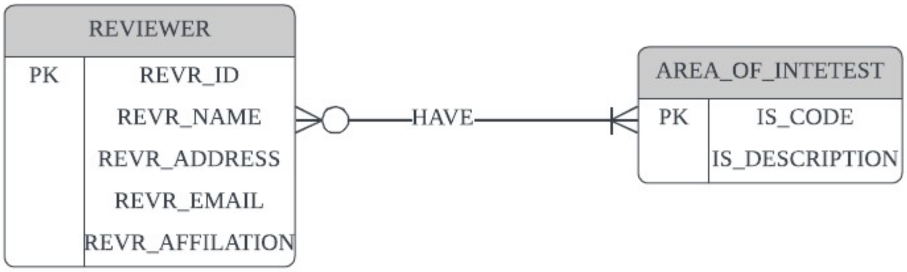
1) **Write**: relation between **SCHOLARS** and **PAPERS**

- One scholar can write one or many papers
- Every paper must have an author or several authors (4)
- M:N (mandatory)



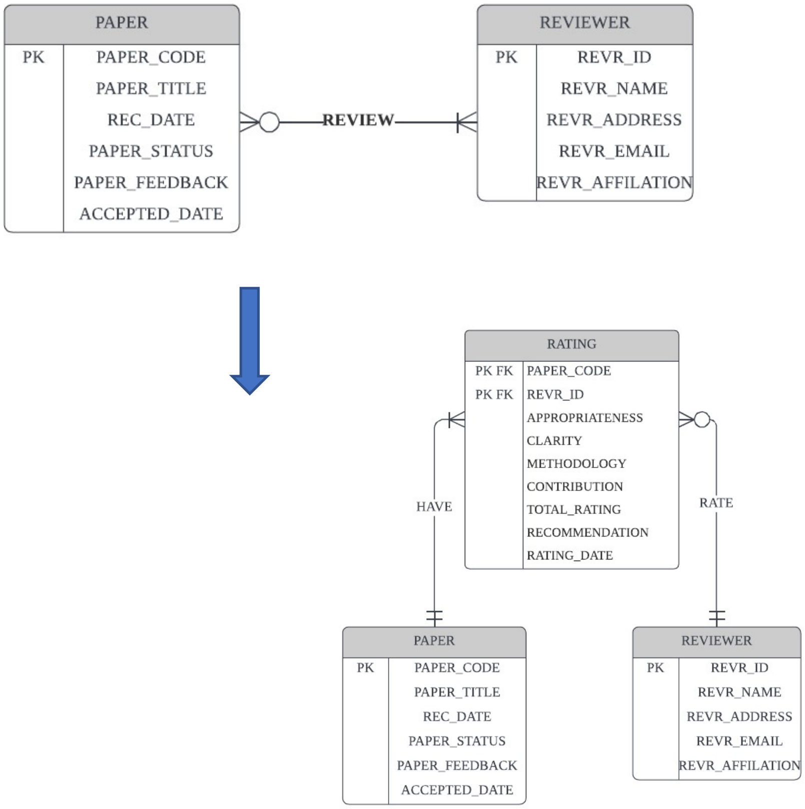
2) Have: relation between **REVIEWERS** and **AREA_OF_INTEREST**

- A reviewer can have many areas of interest, All reviewers must specify **at least one** area of interest. (mandatory)
- An area of interest can be associated with many reviewers. but it is possible to have an area of interest for which the journal has **no reviewers**. (optional)
- M:N



3) **Review:** relation between **REVIEWERS** and **PAPERS**

- Reviewer will typically receive **several** papers to review, new reviewers **may not** have received any paper yet (7) (optional)
- Three or more reviewers will review the paper. (mandatory)
- M:N



PAPER	
PK	PAPER_CODE
	PAPER_TITLE
	REC_DATE
	PAPER_STATUS
	PAPER_FEEDBACK
	ACCEPTED_DATE

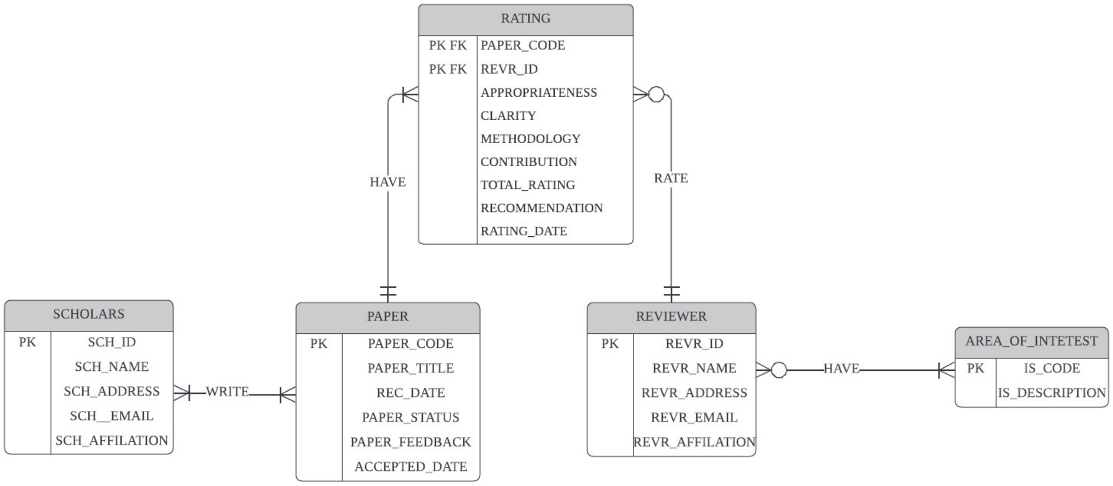
REVIEWER	
PK	REVR_ID
	REVR_NAME
	REVR_ADDRESS
	REVR_EMAIL
	REVR_AFFILATION

RATING	
PK FK	PAPER_CODE
PK FK	REVR_ID
	APPROPRIATENESS
	CLARITY
	METHODOLOGY
	CONTRIBUTION
	TOTAL_RATING
	RECOMMENDATION
	RATING_DATE

PAPER	
PK	PAPER_CODE
	PAPER_TITLE
	REC_DATE
	PAPER_STATUS
	PAPER_FEEDBACK
	ACCEPTED_DATE

REVIEWER	
PK	REVR_ID
	REVR_NAME
	REVR_ADDRESS
	REVR_EMAIL
	REVR_AFFILATION

Step 3: Conceptual ERD



RATING	
PK FK	PAPER_CODE
PK FK	REVR_ID
	APPROPRIATENESS
	CLARITY
	METHODOLOGY
	CONTRIBUTION
	TOTAL_RATING
	RECOMMENDATION
	RATING_DATE

SCHOLARS	
PK	SCH_ID
	SCH_NAME
	SCH_ADDRESS
	SCH__EMAIL
	SCH_AFFILATION

PAPER	
PK	PAPER_CODE
	PAPER_TITLE
	REC_DATE
	PAPER_STATUS
	PAPER_FEEDBACK
	ACCEPTED_DATE

REVIEWER	
PK	REVR_ID
	REVR_NAME
	REVR_ADDRESS
	REVR_EMAIL
	REVR_AFFILATION

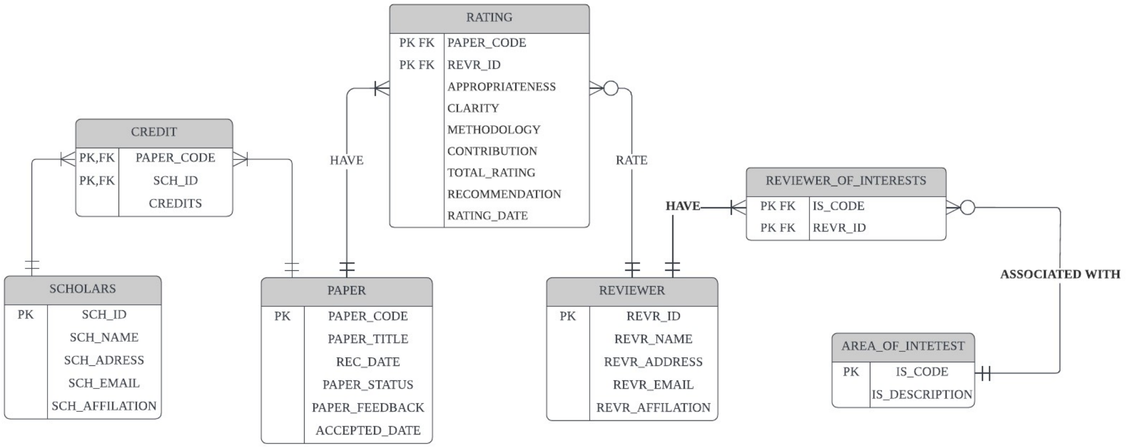
AREA_OF_INTETEST	
PK	IS_CODE
	IS_DESCRIPTION

Logical Database Schema

Process:

1) scholars publish papers, the system will be divided according to the weight of each author in the paper
2) the system will record some basic information of the paper.
3) the editor will change the status of Paper to RECEIVED and select three or more to review the paper according to the interest range of reviewers.
4) Each reviewer will rate the paper after review, and the FEEDBACK is a comprehensive rating of the paper.
5) one paper has three or more RATING, we can set a function to average multiple RATING to get the feedback in PAPER table (because a paper can only have one feedback)
6) the editor can get the range of interest of each reviewer is not in line with the topic of the paper through multiple table query
7) all reviewers' interests are summarized in AREAS_OF_INTERESTS.

The final Crow’s Foot ERD.

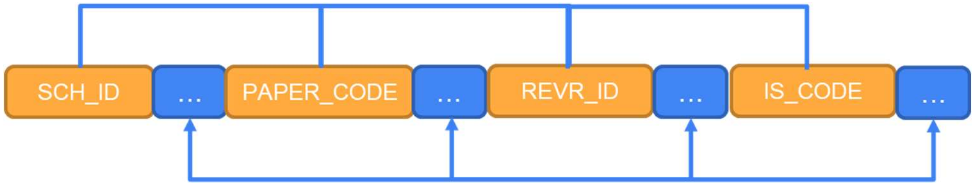


https://lucid.app/lucidchart/9990a63a-0354-49ee-b0e0-203e1ec9f758/edit?invitationId=inv_2774e0c1-44a1-4c6f-9052-2312dd512bb4&page=0_0#

Functional Dependencies and Database Normalization

1NF

1NF(SCH_ID, PAPER_CODE, REVR_ID, IS_CODE,......)



2NF

TABLE: **SCHOLAR**

SCHOLAR (SCH_ID, SCH_NAME,SCH_ADDRESS,SCH_EMAIL,SCH_AFFILATION)

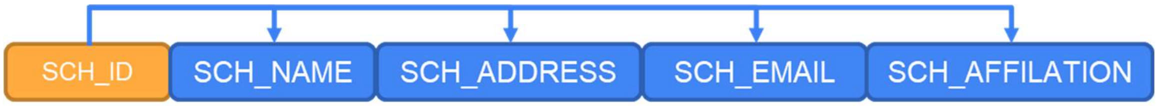


TABLE: **REVIEWER**

REVIEWER (REVR_ID, REVR_NAME, REVR_ADDRESS, REVR_EMAIL, REVR_AFFILATION)

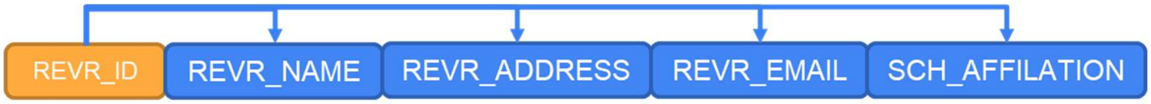


TABLE: **AREA_OF_INTEREST**

AREA_OF_INTEREST (IS_CODE, IS_DISCRIPTION)

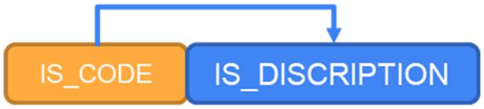
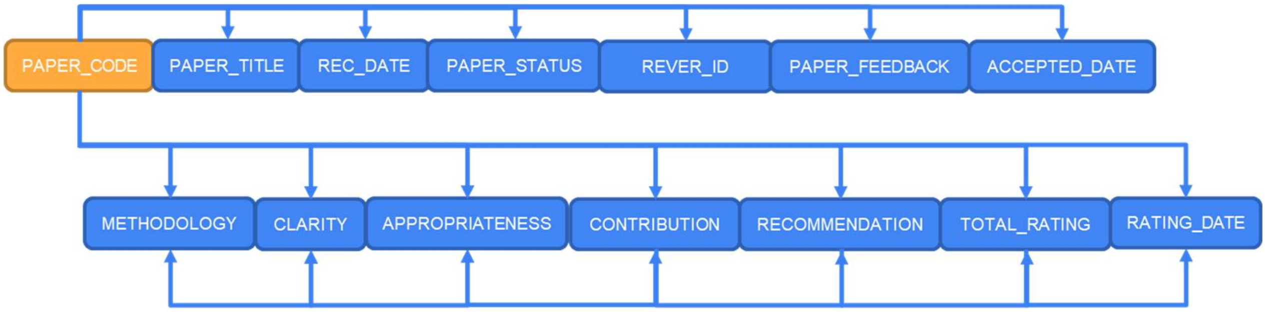


TABLE: **PAPER**

PAPER (PAPER_CODE,PAPER_TITLE,REC_DATE,PAPER_STATUS, ACCEPTED_DATE,APPROPRIATENESS, CLARITY,METHODOLOGY, CONTRIBUTION, RECOMMENDATION, TOTAL_RATING)



Transitive Dependencies

3NF

TABLE: **REVIEWER**

REVIEWER (REVR_ID, REVR_NAME, REVR_ADDRESS, REVR_EMAIL, REVR_AFFILATION)

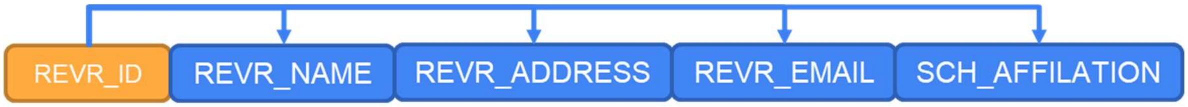


TABLE: **AREA_OF_INTEREST**

AREA_OF_INTEREST (IS_CODE, IS_DISCRIPTION)

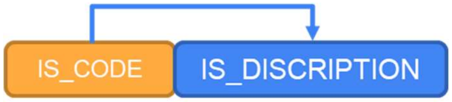


TABLE: **REVIEWER_OF_INTEREST**

REVIEWER_OF_INTEREST(IS_CODE, REVR_ID)

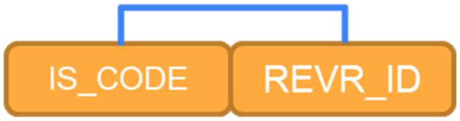


TABLE: **PAPER**

PAPER (PAPER_CODE, PAPER_TITLE, REC_DATE, PAPER_STATUS, PAPER_FEEDBACK, ACCEPTED_DATE)

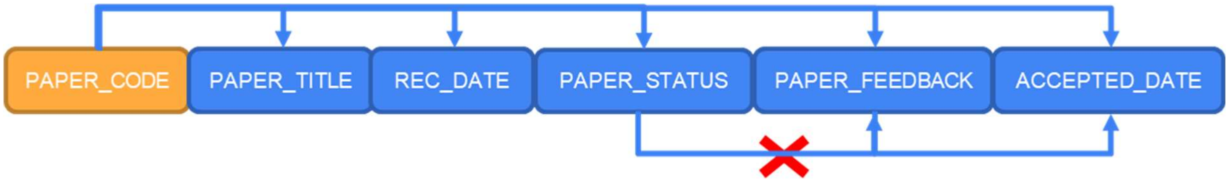


TABLE: **CREDIT**

CREDIT (PAPER_CODE, SCH_ID,CREDIT)

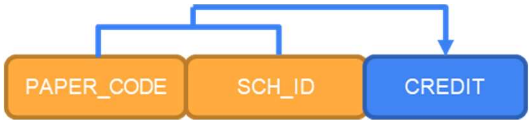


TABLE: **SCHOLAR**

SCHOLAR (SCH_ID, SCH_NAME,SCH_ADDRESS, SCH_EMAIL, SCH_AFFILATION)

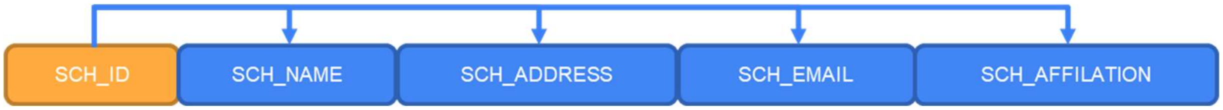


TABLE: **PAPER**

PAPER (PAPER_CODE, PAPER_TITLE, REC_DATE, PAPER_STATUS, PAPER_FEEDBACK, ACCEPTED_DATE)

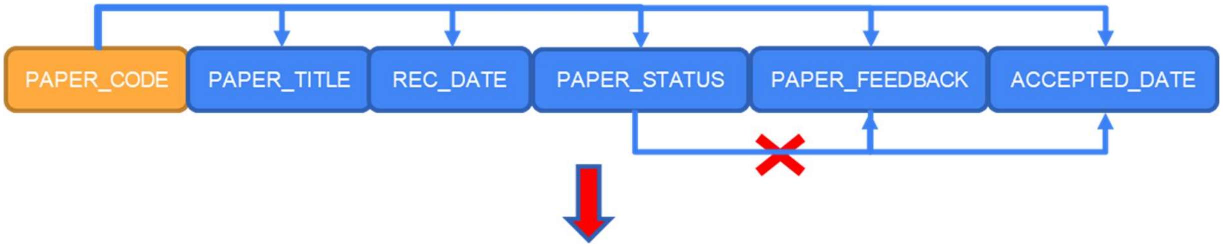


TABLE: **PAPER**

PAPER (PAPER_CODE, PAPER_TITLE, REC_DATE, PAPER_STATUS)

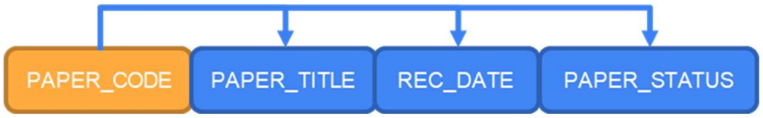


TABLE: **ACCEPTED**

ACCEPTED (PAPER_CODE, PAPER_FEEDBACK, ACCEPTED_DATE)

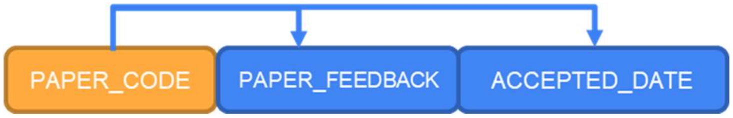
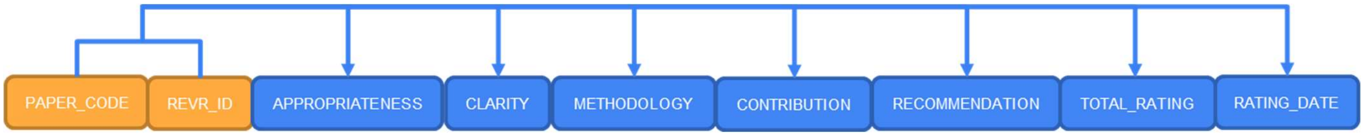


TABLE: **RATING**

RATING (PAPER_CODE, REVR_ID, APPROPRIATENESS, CLARITY, METHODOLOGY, CONTRIBUTION, RECOMMENDATION, TOTAL_RATING, RATING_DATE)



Difficulties

The biggest difficulty of this project is that there is no scope, and there are many things that can be expanded, which will make our team fall into the pursuit of details, thus spending a lot of time in the beginning on discussing the properties of entities.

We later found that our goal was to build a framework that would first enable this database to meet the basic logic and requirements of the project to be implemented, and the specific logic and requirements, we felt, were to continuously improve its functionality and complexity in practice.

The difficulty lies in how to split tables according to the requirements of 3NF, without generating redundant data or unnecessary connections. Combining this problem, how to understand the status of multiple revisions of the paper and only record the date when the paper was accepted is the difficulty of how to split the tables.

The difficulty is that the title implies several sets of many-to-many relationships, which need to be carefully reviewed.

The SQL statements

The SQL statement for each of the defined queries and views, and its corresponding execution results. Please see our submitted script files.