Pulkit Agarwal

19323939

agarwalp@tcd.ie

Abstract

The project below is a relational database I have designed for a MANGA COMICS database.  
It includes the Overview, Entity-Relationship Diagram, Relational Schema, Functional Dependency Diagram and the Source Code along with explanations regarding Normalization, Constraints, Triggers and Security in my Database.

rdbms – manga comics

CSU34041 – INFORMATION MANAGEMENT II

**OVERVIEW**

Manga-Comics refer to a type of graphic-novels emerging from Japan. They can either be hand-drawn or be a computer animation.

People of all ages in Japan read Manga-Comics and has also become quite popular in the western world. These comics come in various genres including comedy, horror, action, adventure, historical, mystery, drama, detective, sports, science fiction and many others.

Manga-Comics are drawn or printed black and white for affordable costs and time constraints. The comics are serialized in magazines with new episodes in every issue and continuing in the next issue.

Artists for these comics typically work alone or with few assistants and are associated with a publishing company. If any of these comics get popular enough, they are typically animated or made into movie during or after its run.

The database I have designed is modelled in the most relational way possible. A few attributes and relationships have been introduced which can be in the tables and diagrams below.

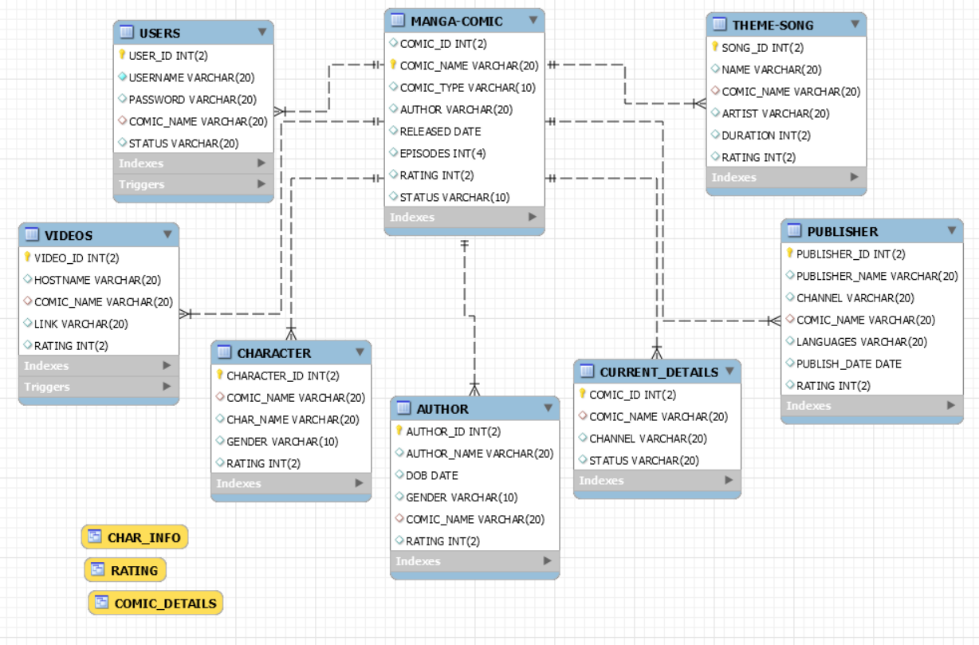
Here is a short description of what each of the tables in the database represent:

1. The table MANGA\_COMIC consists of the name of the comic, its type, its author, its rating and the number of episodes each comic contains.
2. The table AUTHOR consists of the name of the author, their date of birth, their gender, the name of the comic they have produced and their personal rating.
3. The table CHARACTERS comprise of the name of the character along with the show they appear in, their gender and their character rating.
4. The table PUBLISHER consists of all the information about the publisher’s name, the channel on which the show appears, the languages in which the show is available, the link to the channel, the name of the comic and the publisher’s rating.
5. The table THEME\_SONG includes the name of the theme song, the artist who performs in the theme song, the name of the comic for the song, the duration and rating of the song.
6. The table SHOW\_DETAILS includes the show details such as the comic name, the channel on which the show is broadcasted and the current story summary of the show.
7. The table VIDEOS consists of the name of the server where the videos are stored along with their comic name, it also includes the link to the channel where videos can be viewed and the rating on the basis of the availability and quality of videos for a comic.
8. The table USERS comprise of the username and password of a user, it also includes the comic status of the comic the user is currently watching or will-watch or has watched. Lastly, it has a rating column to record the ratings a particular user has given to the comic.

**ENTITY-RELATIONSHIP DIAGRAM**

**Diagram

Description automatically generated**

**RELATIONAL-SCHEMA DIAGRAM**

**FUNCTIONAL-DEPENDENCY DIAGRAM**

**NORMALISATION:**

The tables that I have created for my database were already normalised. Additionally, I added 2 tables with composite primary keys to model the many-to-many-relationships and using that the many-to-many attributes have been normalised.

**SEMANTIC CONSTRAINTS:**

I have added various semantic constraints to the database I have designed for this project.

The first constraints I introduced were for checking particular words in a column.

For example:

CONSTRAINT COMIC\_TYPE CHECK ((TYPE = 'TV-SERIES') OR (TYPE = 'MOVIE') OR (TYPE = 'OVA'))

CONSTRAINT AUTHOR\_GENDER CHECK ((GENDER = 'MALE') OR (GENDER = 'FEMALE'))

CONSTRAINT CHECK­\_STATUS CHECK ((STATUS = 'WILL-WATCH') OR (STATUS = 'WATCHED') OR STATUS = 'WATCHING'))

I also introduced numerical constraints to almost every table in the database. This will prevent the value of a certain variable in the table to be less than the constrained value.

For example:

CONSTRAINT COMIC\_RATING CHECK (RATING <= 10.00)

CONSTRAINT SONG\_LENGTH CHECK (DURATION <= 2.00)

Apart from these, the other constraints that I have introduced into my database are NOT NULL constraint for most of the entries in the tables. And most importantly, I have used PRIMARY KEY and FOREIGN KEY constraints in my database to prevent duplicate entries and manage efficiency.

**TRIGGERS:**

I have introduced three triggers in my database. One of the examples of these triggers is shown below:

CREATE TRIGGER NEW\_COMIC AFTER INSERT ON MANGA\_COMIC

FOR EACH ROW

BEGIN

INSERT INTO SHOW\_DETAILS SET COMIC\_NAME = NEW.COMIC\_NAME ;

INSERT INTO AUTHOR SET COMIC\_NAME = NEW.COMIC\_NAME ;

END;

In this particular trigger, whenever a new comic is introduced in the MANGA\_COMIC table, the trigger inserts a new comic name in the SHOW\_DETAILS and the AUTHOR tables with that particular new comic name.

**SECURITY:**

For adding security in my Manga Comic Database, I have introduced the following methods:

1. I have created three roles for my database.

CREATE ROLE 'DB\_developer', 'DB\_read', 'DB\_write';

1. I have granted certain permissions for those certain roles depending on their roles.

GRANT ALL ON mangacomicdb.\* TO 'DB\_developer';

GRANT SELECT ON mangacomicdb.\* TO 'DB\_read';

GRANT INSERT, UPDATE, DELETE ON mangacomicdb.\* TO 'DB\_write';

1. I have then created users along with their passwords to access the database.

CREATE USER 'developer1'@'localhost' IDENTIFIED BY 'developer1pass';

CREATE USER 'read\_dev1'@'localhost' IDENTIFIED BY 'read\_dev1pass';

CREATE USER 'read\_dev2'@'localhost' IDENTIFIED BY 'read\_dev2pass';

1. Finally, the users introduced are granted the earlier defined roles to access and manipulate the database.

GRANT 'DB\_developer' TO 'developer1'@'localhost';

GRANT 'DB\_read' TO 'read\_dev1'@'localhost', 'read\_dev2'@'localhost';

**APPENDIX**

CREATE TABLE MANGA\_COMIC (

COMIC\_ID INT,

COMIC\_NAME VARCHAR(20),

TYPE VARCHAR(10),

AUTHOR VARCHAR(20),

RELEASED DATE,

EPISODES INT,

RATING INT,

COMIC\_STATUS VARCHAR(10),

CONSTRAINT COMIC\_RATING CHECK (RATING <= 10.00),

CONSTRAINT COMIC\_TYPE CHECK ((TYPE = 'TV-SERIES') OR (TYPE = 'MOVIE')

OR (TYPE = 'OVA')),

CONSTRAINT MANGA\_COMIC\_PK PRIMARY KEY (COMIC\_NAME)

);

CREATE TABLE AUTHOR (

AUTHOR\_ID INT,

AUTHOR\_NAME VARCHAR(20),

DOB DATE,

GENDER VARCHAR(10),

COMIC\_NAME VARCHAR(20),

RATING INT,

CONSTRAINT AUTHOR\_RATING CHECK (RATING <= 10.00),

CONSTRAINT AUTHOR\_GENDER CHECK ((GENDER = 'MALE') OR (GENDER = 'FEMALE')),

CONSTRAINT AUTHOR\_PK PRIMARY KEY (AUTHOR\_ID)

);

CREATE TABLE CHARACTERS (

CHARACTER\_ID INT,

CHAR\_NAME VARCHAR(20),

COMIC\_NAME VARCHAR(20),

GENDER VARCHAR(10),

RATING INT,

CONSTRAINT CHAR\_RATING CHECK (RATING <= 10.00),

CONSTRAINT CHAR\_GENDER CHECK ((GENDER = 'MALE') OR (GENDER = 'FEMALE')),

CONSTRAINT CHARACTER\_PK PRIMARY KEY (CHARACTER\_ID)

);

CREATE TABLE THEME\_SONG (

SONG\_ID INT,

NAME VARCHAR(20),

ARTIST VARCHAR(20),

COMIC\_NAME VARCHAR(20),

DURATION INT,

RATING INT,

CONSTRAINT SONG\_RATING CHECK (RATING <= 10.00),

CONSTRAINT SONG\_LENGTH CHECK (DURATION <= 2.00),

CONSTRAINT SONG\_PK PRIMARY KEY (SONG\_ID)

);

CREATE TABLE SHOW\_DETAILS (

COMIC\_ID INT,

COMIC\_NAME VARCHAR(20),

CHANNEL\_NAME VARCHAR(20),

DETAILS VARCHAR(30),

CONSTRAINT SHOW\_DETAILS\_PK PRIMARY KEY (COMIC\_ID)

);

CREATE TABLE PUBLISHER (

PUBLISHER\_ID INT,

PUBLISHER\_NAME VARCHAR(20),

CHANNEL\_NAME VARCHAR(20),

COMIC\_NAME VARCHAR(20),

LANGUAGES VARCHAR(20),

LINK VARCHAR(20),

RATING INT,

CONSTRAINT P\_RATING CHECK (RATING <= 10.00),

CONSTRAINT PUBLISHER\_PK PRIMARY KEY (PUBLISHER\_ID)

);

CREATE TABLE VIDEOS (

VIDEO\_ID INT,

HOSTNAME VARCHAR(20),

COMIC\_NAME VARCHAR(20),

LINK VARCHAR(20),

RATING INT,

CONSTRAINT VIDEO\_RATING CHECK (RATING <= 10.00),

CONSTRAINT VIDEOS\_PK PRIMARY KEY (VIDEO\_ID)

);

CREATE TABLE USERS (

USER\_ID INT,

USERNAME VARCHAR(20),

PASSWORD VARCHAR(20),

COMIC\_NAME VARCHAR(20),

STATUS VARCHAR(20),

RATING INT,

CONSTRAINT USER\_RATING CHECK (RATING <= 10.00),

CONSTRAINT CHECK­\_STATUS CHECK ((STATUS = 'WILL-WATCH')

OR (STATUS = 'WATCHED')

OR (STATUS = 'WATCHING')),

CONSTRAINT USER\_ID PRIMARY KEY (USER\_ID)

);

SELECT \* FROM MANGA\_COMIC;

SELECT \* FROM CHARACTERS;

SELECT \* FROM AUTHOR;

SELECT \* FROM PUBLISHER;

SELECT \* FROM SHOW\_DETAILS;

SELECT \* FROM THEME\_SONG;

SELECT \* FROM USERS;

SELECT \* FROM VIDEOS;

ALTER TABLE CHARACTERS

ADD CONSTRAINT MANGA\_COMIC\_FK

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

ALTER TABLE AUTHOR

ADD CONSTRAINT MANGA\_COMIC\_FK1

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

ALTER TABLE PUBLISHER

ADD CONSTRAINT MANGA\_COMIC\_FK2

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

ALTER TABLE SHOW\_DETAILS

ADD CONSTRAINT MANGA\_COMIC\_FK3

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

ALTER TABLE THEME\_SONG

ADD CONSTRAINT MANGA\_COMIC\_FK4

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

ALTER TABLE USERS

ADD CONSTRAINT MANGA\_COMIC\_FK5

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

ALTER TABLE VIDEOS

ADD CONSTRAINT MANGA\_COMIC\_FK6

FOREIGN KEY(COMIC\_NAME)

REFERENCES MANGA\_COMIC(COMIC\_NAME);

INSERT INTO MANGA\_COMIC VALUES(1, 'NARUTO-ORIGINAL', 'TV-SERIES', 'Masashi Kishimoto', '2002-10-04', 220, 9, 'COMPLETED');

INSERT INTO MANGA\_COMIC VALUES(2, 'NARUTO-SHIPPUDEN', 'TV-SERIES', 'Masashi Kishimoto', '2007-02-16', 500, 10, 'COMPLETED');

INSERT INTO MANGA\_COMIC VALUES(3, 'BORUTO: THE MOVIE OF NARUTO', 'MOVIE', 'UKYO KODACHI', '2015-08-08', 4, 9, 'COMPLETED');

INSERT INTO MANGA\_COMIC VALUES(4, 'BORUTO: NEXT GENERATIONS', 'TV-SERIES', 'UKYO KODACHI', '2017-04-06', 83, 9, 'ON-GOING');

INSERT INTO MANGA\_COMIC VALUES(5, 'The Day Naruto Became Hokage', 'OVA', 'Masashi Kishimoto', '2016-05-05', 1, 9, 'COMPLETED');

INSERT INTO MANGA\_COMIC VALUES(6, 'BOKU NO HERO ACADEMIA', 'TV-SERIES', 'Kohei Horikoshi', '2016-05-05', 75, 10, 'ON-GOING');

INSERT INTO MANGA\_COMIC VALUES(7, 'KIMI NO-NA-WA', 'MOVIE', 'MAKATO SHINKAI', '2016-05-04', 1, 10, 'COMPLETED');

INSERT INTO CHARACTERS VALUES( 1, 'NARUTO', 'NARUTO-ORIGINAL', 'MALE', 9.63);

INSERT INTO CHARACTERS VALUES( 2, 'KAKASHI', 'NARUTO-ORIGINAL', 'MALE', 8.97);

INSERT INTO CHARACTERS VALUES( 3, 'SAKURA', 'NARUTO-ORIGINAL', 'FEMALE', 8.77);

INSERT INTO CHARACTERS VALUES( 4, 'YAMATO', 'NARUTO-ORIGINAL', 'MALE', 8.33);

INSERT INTO CHARACTERS VALUES( 5, 'SASUKE', 'NARUTO-ORIGINAL', 'MALE', 8.13);

INSERT INTO AUTHOR VALUES(1, 'Kohei Horikoshi', '1987-07-09', 'MALE', 'BOKU NO HERO ACADEMIA', 9.14);

INSERT INTO AUTHOR VALUES(2, 'Masashi Kishimoto', '1974-11-08', 'MALE', 'NARUTO-ORIGINAL', 7.65);

INSERT INTO AUTHOR VALUES(3, 'MAKATO SHINKAI', '1983-02-19', 'MALE', 'KIMI NO-NA-WA', 10);

INSERT INTO AUTHOR VALUES(4, 'UKYO KODACHI', '1979-04-01', 'MALE', 'BORUTO: NEXT GENERATIONS', 7.61);

INSERT INTO AUTHOR VALUES(5, 'Masashi Kishimoto', '1974-11-08', 'MALE', 'NARUTO-SHIPPUDEN', 7.63);

INSERT INTO THEME\_SONG VALUES(1, 'GO ON', 'THE TEANGENT','KIMI NO-NA-WA', 1.57, 9.6);

INSERT INTO THEME\_SONG VALUES(2, 'No Boy,No Cry', 'Stance Punks', 'NARUTO-ORIGINAL', 1.24, 7.65);

INSERT INTO THEME\_SONG VALUES(3, 'DIVER', 'ELENA','NARUTO-SHIPPUDEN', 1.59, 8.21);

INSERT INTO THEME\_SONG VALUES(4, 'Lonely Go!', 'Brian the Sun', 'BORUTO: NEXT GENERATIONS', 2.30, 7.53);

INSERT INTO THEME\_SONG VALUES(5, 'BLUE BIRD', 'Ikimono Gakari','NARUTO-SHIPPUDEN', 3.35, 7.73);

INSERT INTO SHOW\_DETAILS VALUES(1, 'NARUTO-ORIGINAL', 'TV-TOKYO', 'Naruto trains for the competition at the academy...');

INSERT INTO SHOW\_DETAILS VALUES(2, 'NARUTO-SHIPPUDEN', 'TV-TOKYO', 'Naruto faces his biggest rival for first time...');

INSERT INTO SHOW\_DETAILS VALUES(3, 'BOKU NO HERO ACADEMIA', 'YOUPEI', 'The team conquers new war academy...');

INSERT INTO SHOW\_DETAILS VALUES(4, 'KIMI NO-NA-WA', 'YOUPEI', '');

INSERT INTO SHOW\_DETAILS VALUES(5, 'BORUTO: NEXT GENERATIONS', 'TV-TOKYO', 'The Adventures Continues... ');

INSERT INTO PUBLISHER VALUES(1, 'VIZ-MEDIA', 'TV-TOKYO', 'NARUTO-SHIPPUDEN', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',8.34);

INSERT INTO PUBLISHER VALUES(2, 'VIZ-MEDIA', 'TV-TOKYO', 'KIMI NO-NA-WA', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',9.88);

INSERT INTO PUBLISHER VALUES(3, 'VIZ-MEDIA', 'TV-TOKYO', 'BOKU NO HERO ACADEMIA', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',9.10);

INSERT INTO PUBLISHER VALUES(4, 'VIZ-MEDIA', 'TV-TOKYO', 'NARUTO-ORIGINAL', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',7.21);

INSERT INTO PUBLISHER VALUES(5, 'VIZ-MEDIA', 'TV-TOKYO', 'BORUTO: NEXT GENERATIONS', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM', 7.89);

INSERT INTO VIDEOS VALUES(1, 'RAPIDFIRE', 'NARUTO-ORIGINAL', 'WWW.GOGO-ANIME.COM', 7.21);

INSERT INTO VIDEOS VALUES(2, 'RAP', 'KIMI NO-NA-WA', 'WWW.GOGO-ANIME.COM' ,9.88);

INSERT INTO VIDEOS VALUES(3, 'FIRE', 'BOKU NO HERO ACADEMIA', 'WWW.GOGO-ANIME.COM', 9.10);

INSERT INTO VIDEOS VALUES(4, 'RAPIDFIRE', 'BORUTO: NEXT GENERATIONS', 'WWW.GOGO-ANIME.COM', 7.89);

INSERT INTO VIDEOS VALUES(5, 'RARE', 'NARUTO-SHIPPUDEN', 'WWW.GOGO-ANIME.COM', 8.34);

INSERT INTO USERS VALUES(1, 'JERRY','12345','NARUTO-ORIGINAL','WATCHING',8.08);

INSERT INTO USERS VALUES(2,'RICK','12345', 'BORUTO: NEXT GENERATIONS','WATCHING',7.45);

INSERT INTO USERS VALUES(3, 'MORTY','00000','NARUTO-SHIPPUDEN','WILL-WATCH',7.31);

INSERT INTO USERS VALUES(4, 'TOM','TOM123','KIMI NO-NA-WA','WATCHED',10);

INSERT INTO USERS VALUES(5, 'JERRY','QWERTY','BOKU NO HERO ACADEMIA','WATCHING',9.43);

delimiter |

CREATE TRIGGER DEL\_USER BEFORE DELETE ON USERS

FOR EACH ROW

BEGIN

DELETE FROM AUTHOR WHERE COMIC\_NAME = USERS.COMIC\_NAME;

END;

|

delimiter ;

delimiter |

CREATE TRIGGER NEW\_COMIC AFTER INSERT ON MANGA\_COMIC

FOR EACH ROW

BEGIN

INSERT INTO SHOW\_DETAILS SET COMIC\_NAME = NEW.COMIC\_NAME ;

INSERT INTO AUTHOR SET COMIC\_NAME = NEW.COMIC\_NAME ;

END;

|

delimiter ;

delimiter |

CREATE TRIGGER NEW\_USER AFTER INSERT ON VIDEOS

FOR EACH ROW

BEGIN

INSERT INTO USERS SET COMIC\_NAME = NEW.COMIC\_NAME ;

INSERT INTO USERS SET USERNAME = NEW.HOSTNAME ;

END;

|

delimiter ;

CREATE VIEW RATING AS SELECT COMIC\_ID, COMIC\_NAME, AUTHOR, RATING FROM MANGA\_COMIC;

CREATE VIEW CHAR\_INFO AS SELECT CHARACTER\_ID, CHAR\_NAME, GENDER, COMIC\_NAME, RATING FROM CHARACTERS;

CREATE VIEW COMIC\_DETAILS AS SELECT PUBLISHER\_ID, PUBLISHER\_NAME, CHANNEL\_NAME, COMIC\_NAME, LINK, RATING FROM PUBLISHER;

CREATE ROLE 'DB\_developer', 'DB\_read', 'DB\_write';

GRANT ALL ON mangacomicdb.\* TO 'DB\_developer';

GRANT SELECT ON mangacomicdb.\* TO 'DB\_read';

GRANT INSERT, UPDATE, DELETE ON mangacomicdb.\* TO 'DB\_write';

CREATE USER 'developer1'@'localhost' IDENTIFIED BY 'developer1pass';

CREATE USER 'read\_dev1'@'localhost' IDENTIFIED BY 'read\_dev1pass';

CREATE USER 'read\_dev2'@'localhost' IDENTIFIED BY 'read\_dev2pass';

GRANT 'DB\_developer' TO 'developer1'@'localhost';

GRANT 'DB\_read' TO 'read\_dev1'@'localhost', 'read\_dev2'@'localhost';

DROP TABLE CHARACTERS;

DROP TABLE AUTHOR;

DROP TABLE PUBLISHER;

DROP TABLE SHOW\_DETAILS;

DROP TABLE THEME\_SONG;

DROP TABLE USERS;

DROP TABLE VIDEOS;

DROP TABLE MANGA\_COMIC;