RDBMS – MANGA COMICS

CSU34041 – INFORMATION MANAGEMENT II

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.

Pulkit Agarwal
19323939
agarwalp@tcd.ie

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.

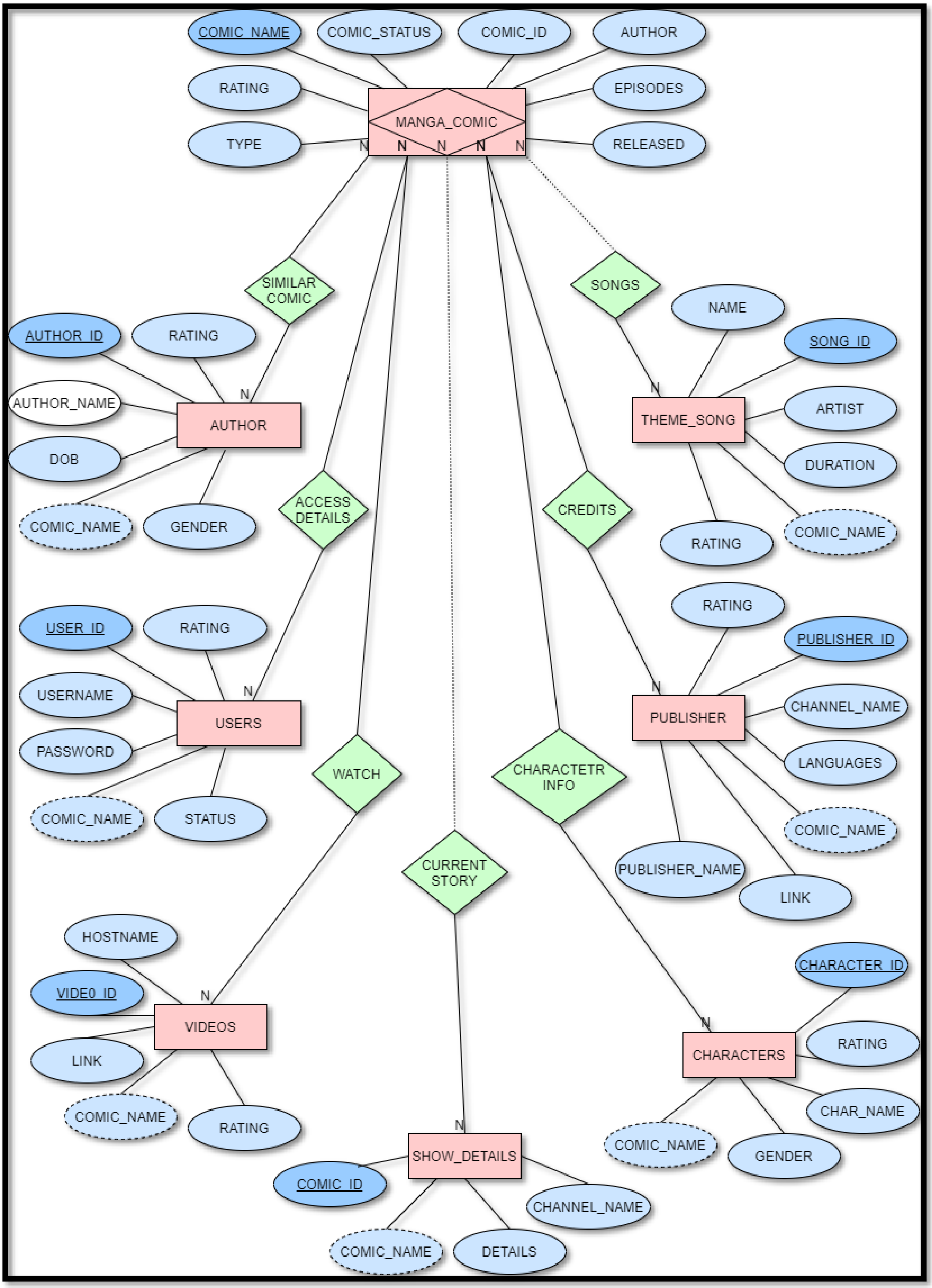
5. The table THEME_SONG includes the name of the theme song, the artist who performs in the theme song, the name of the comic for the song, the duration and rating of the song.

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.

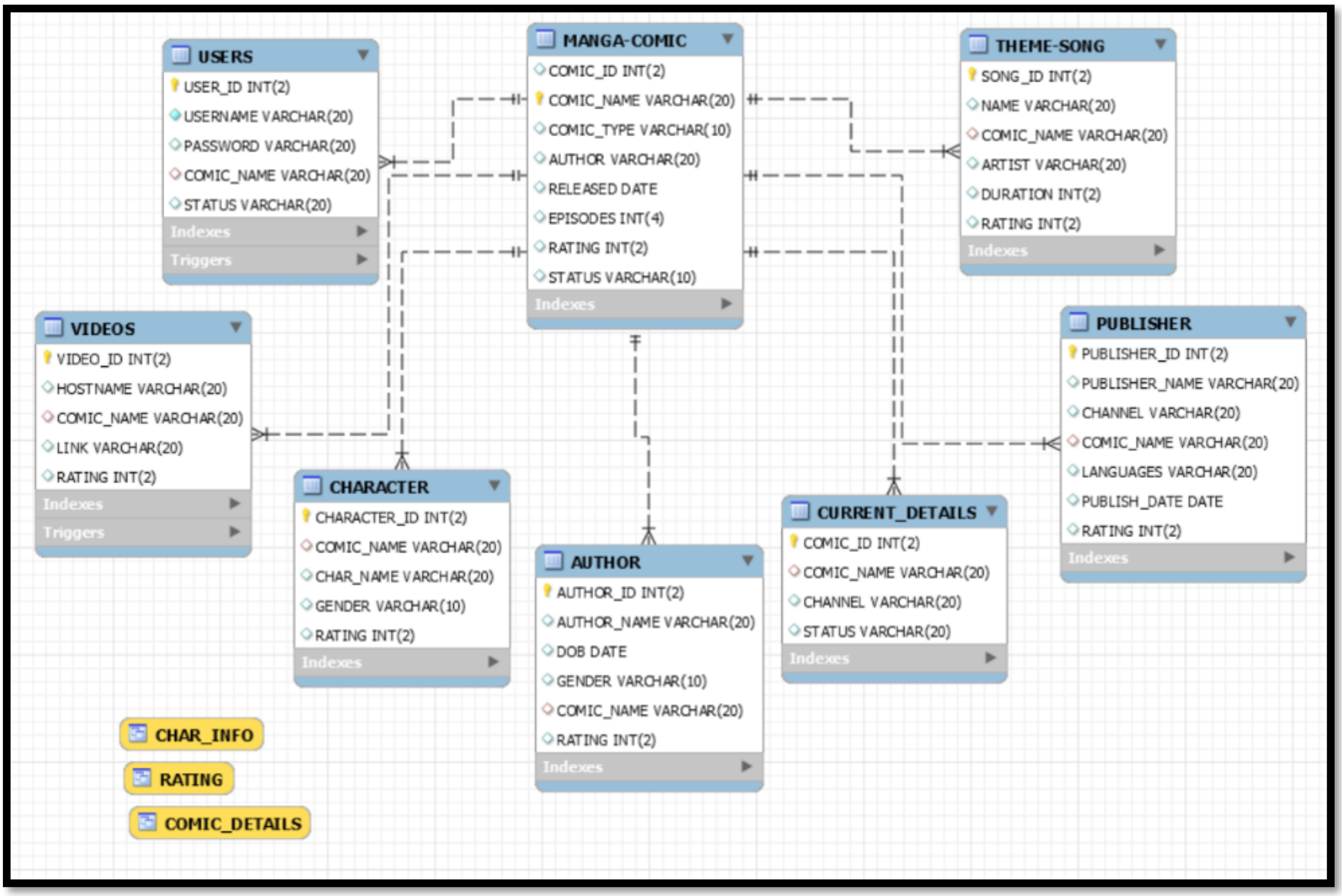
7. The table VIDEOS consists of the name of the server where the videos are stored along with their comic name, it also includes the link to the channel where videos can be viewed and the rating on the basis of the availability and quality of videos for a comic.

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.

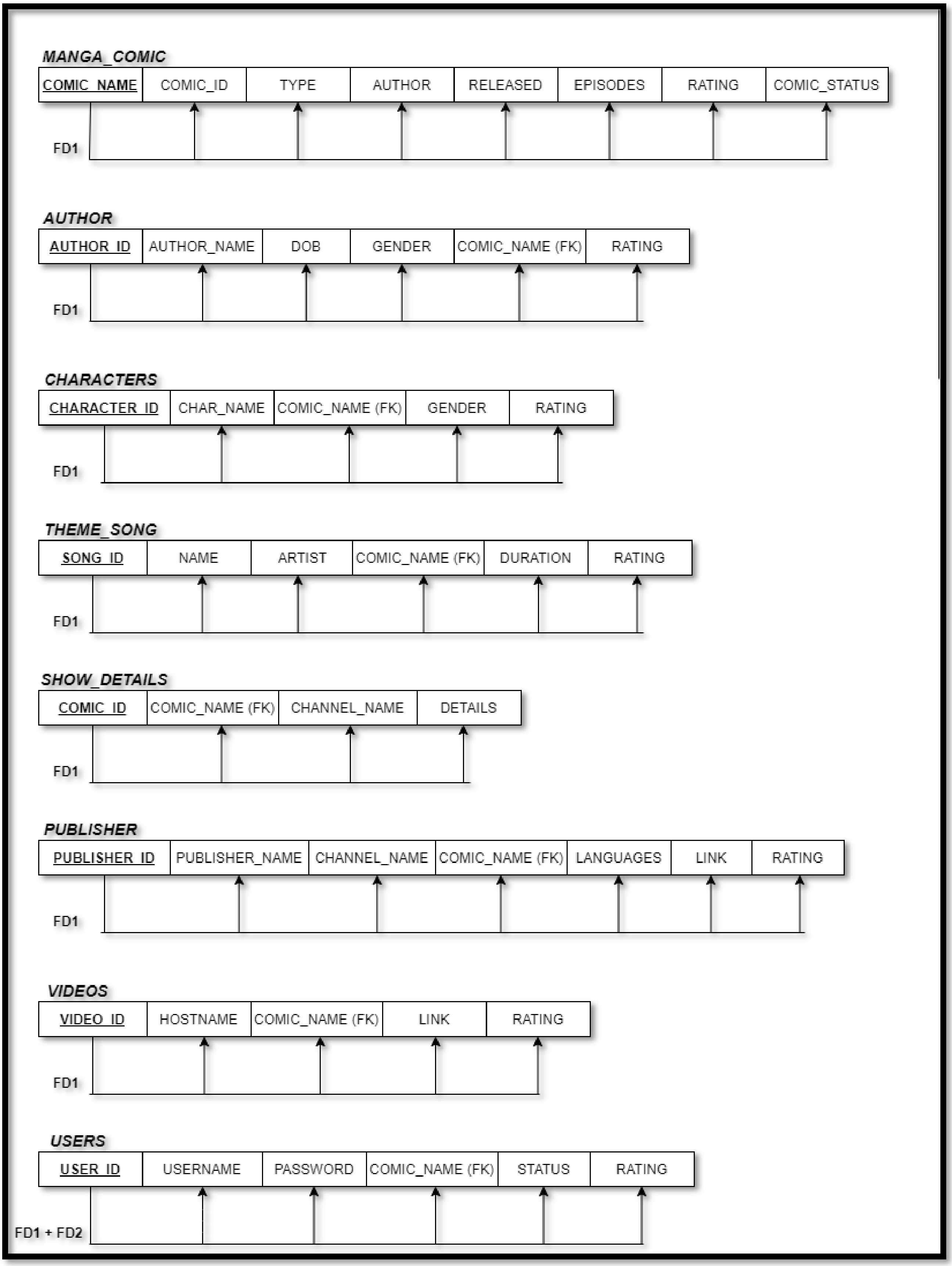
ENTITY-RELATIONSHIP DIAGRAM



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 AUTHOR_GENDER CHECK ((GENDER = 'MALE') OR (GENDER = 'FEMALE'))
CONSTRAINT COMIC_TYPE CHECK ((TYPE = 'EPISODES') OR (TYPE = 'PICTURE') OR (TYPE = 'COMIC')),
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:

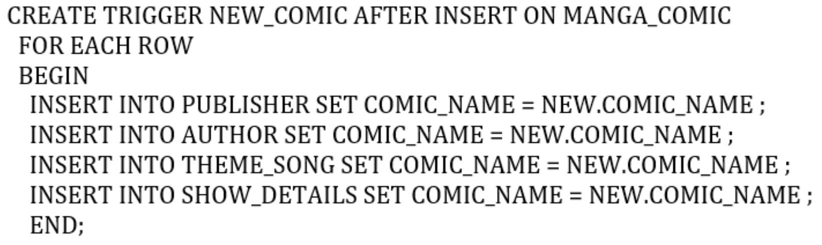
```
CONSTRAINT COMIC_RATING CHECK (RATING <= 10.00)

CONSTRAINT SONG_LENGTH CHECK (DURATION <= 2.00)
```

I have used PRIMARY KEY and FOREIGN KEY constraints in my database to prevent duplicate entries and manage efficiency and NOT NULL constraints in every entity to prevent empty cells in tables.

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 PUBLISHER SET COMIC_NAME = NEW.COMIC_NAME ;
   INSERT INTO AUTHOR SET COMIC_NAME = NEW.COMIC_NAME ;
   INSERT INTO THEME_SONG SET COMIC_NAME = NEW.COMIC_NAME ;
   INSERT INTO SHOW_DETAILS 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';
```

2. 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';
```

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

```
CREATE USER 'databaseDEV01'@'localhost' IDENTIFIED BY 'DEV1pass';
CREATE USER 'read_dev1'@'localhost' IDENTIFIED BY 'read_dev1pass';
CREATE USER 'read_dev2'@'localhost' IDENTIFIED BY 'read_dev2pass';
```

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

```
GRANT 'DB_developer' TO 'databaseDEV01'@'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 = 'EPISODES') OR (TYPE = 'PICTURE')
      OR (TYPE = 'COMIC')),
   CONSTRAINT MANGA_COMIC_PK PRIMARY KEY (COMIC_NAME)
);


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)
);


CREATE TABLE AUTHOR (
   AUTHOR_ID INT,
   AUTHOR_NAME VARCHAR(20),
   DATE_OF_BIRTH 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 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 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 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)
);


SELECT * FROM MANGA_COMIC;
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, 'DEATH NOTE', 'EPISODES', 'YOSHINAGA', '2004-11-14', 220, 9, 'FINISHED');
INSERT INTO MANGA_COMIC VALUES(2, 'BLUE EXORCIST', 'EPISODES', 'YOSHINAGA', '2005-01-26', 500, 10, 'FINISHED');
INSERT INTO MANGA_COMIC VALUES(3, 'AKIRA', 'PICTURE', 'MILINO SAWA', '2013-06-07', 4, 9, 'FINISHED');
INSERT INTO MANGA_COMIC VALUES(4, 'FULLMETAL-ALCHEMIST', 'EPISODES', 'MILINO SAWA', '2016-09-26', 83, 9, 'RUNNING');
INSERT INTO MANGA_COMIC VALUES(5, 'The Day Naruto Became Hokage', 'COMIC', 'YOSHINAGA', '2006-09-25', 1, 9, 'FINISHED');
INSERT INTO MANGA_COMIC VALUES(6, 'KOCHIKAME', 'EPISODES', 'HARISHIWA CHAN', '2015-08-13', 75, 10, 'RUNNING');
INSERT INTO MANGA_COMIC VALUES(7, 'DRAGONBALL', 'PICTURE', 'MAKATO SHINKAI', '2014-07-21', 1, 10, 'FINISHED');
INSERT INTO USERS VALUES(1, 'JERRY','12345','DEATH NOTE','WATCHING',8.08);
INSERT INTO USERS VALUES(2,'RICK987','12345abcd', 'FULLMETAL-ALCHEMIST','WATCHING',7.45);
INSERT INTO USERS VALUES(3, 'MORTY','00000','BLUE EXORCIST','WILL-WATCH',7.31);
INSERT INTO USERS VALUES(4, 'TOMboy','TOMxxx123','DRAGONBALL','WATCHED',10);
INSERT INTO USERS VALUES(5, 'JERRY','QWERTY','KOCHIKAME','WATCHING',9.43);


INSERT INTO PUBLISHER VALUES(1, 'VIZ-MEDIA', 'TV-TOKYO', 'BLUE EXORCIST', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',8.34);
INSERT INTO PUBLISHER VALUES(2, 'VIZ-MEDIA', 'TV-TOKYO', 'DRAGONBALL', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',9.88);
INSERT INTO PUBLISHER VALUES(3, 'VIZ-MEDIA', 'TV-TOKYO', 'KOCHIKAME', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',9.10);
INSERT INTO PUBLISHER VALUES(4, 'VIZ-MEDIA', 'TV-TOKYO', 'DEATH NOTE', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM',7.21);
INSERT INTO PUBLISHER VALUES(5, 'VIZ-MEDIA', 'TV-TOKYO', 'FULLMETAL-ALCHEMIST', 'ENGLISH, JAPANESE', 'WWW.GOGO-ANIME.COM', 7.89);


INSERT INTO THEME_SONG VALUES(1, 'LET ME GO', 'HIROSHI YOSHIDA','DRAGONBALL', 1.57, 9.6);
INSERT INTO THEME_SONG VALUES(2, 'DO NOY CRY BABY', 'TOSHI YOSHIDA','DEATH NOTE', 1.24, 7.65);
INSERT INTO THEME_SONG VALUES(3, 'EXCORCIST', 'SHINSUI ITO','BLUE EXORCIST', 1.59, 8.21);
INSERT INTO THEME_SONG VALUES(4, 'DILEMMA', 'OHARA KOSON','FULLMETAL-ALCHEMIST', 2.30, 7.53);
INSERT INTO THEME_SONG VALUES(5, 'BABY BLUE SEA', 'TOSHI YOSHIDA','BLUE EXORCIST', 3.35, 7.73);


INSERT INTO SHOW_DETAILS VALUES(1, 'DEATH NOTE', 'TV-TOKYO', 'Naruto trains for the competition at the academy...');
INSERT INTO SHOW_DETAILS VALUES(2, 'BLUE EXORCIST', 'TV-TOKYO', 'Naruto faces his biggest rival for first time...');
INSERT INTO SHOW_DETAILS VALUES(3, 'KOCHIKAME', 'PINNAI', 'The team conquers new war academy...');
INSERT INTO SHOW_DETAILS VALUES(4, 'DRAGONBALL', 'PINNAI', 'Konnichiwa');
INSERT INTO SHOW_DETAILS VALUES(5, 'FULLMETAL-ALCHEMIST', 'TV-TOKYO', 'The Adventures Continues... ');


INSERT INTO AUTHOR VALUES(1, 'HARISHIWA CHAN', '1987-07-09', 'MALE', 'KOCHIKAME', 9.14);
INSERT INTO AUTHOR VALUES(2, 'YOSHINAGA', '1974-11-08', 'MALE', 'DEATH NOTE', 7.65);
INSERT INTO AUTHOR VALUES(3, 'MAKATO SHINOWA', '1983-02-19', 'MALE', 'DRAGONBALL', 10);
INSERT INTO AUTHOR VALUES(4, 'MILINO SAWA', '1979-04-01', 'MALE', 'FULLMETAL-ALCHEMIST', 7.61);
INSERT INTO AUTHOR VALUES(5, 'YOSHINAGA', '1974-11-08', 'MALE', 'BLUE EXORCIST', 7.63);


INSERT INTO VIDEOS VALUES(1, 'RAPIDFIRE', 'DEATH NOTE', 'WWW.GOGO-ANIME.COM', 7.21);
INSERT INTO VIDEOS VALUES(2, 'RAP', 'DRAGONBALL', 'WWW.GOGO-ANIME.COM' ,9.88);
INSERT INTO VIDEOS VALUES(3, 'RAPIDFIRE', 'FULLMETAL-ALCHEMIST', 'WWW.GOGO-ANIME.COM', 7.89);
INSERT INTO VIDEOS VALUES(4, 'RARE', 'BLUE EXORCIST', 'WWW.GOGO-ANIME.COM', 8.34);
INSERT INTO CHARACTERS VALUES( 1, 'RYUK', 'DEATH NOTE', 'MALE', 9.63);
INSERT INTO CHARACTERS VALUES( 2, 'L', 'DEATH NOTE', 'MALE', 8.97);
INSERT INTO CHARACTERS VALUES( 3, 'MISA AMANE', 'DEATH NOTE', 'FEMALE', 8.77);
INSERT INTO CHARACTERS VALUES( 4, 'NEAR', 'DEATH NOTE', 'MALE', 8.33);
INSERT INTO CHARACTERS VALUES( 5, 'MELLO', 'DEATH NOTE', 'MALE', 8.13);


delimiter |
CREATE TRIGGER USER_LEFT BEFORE DELETE ON USERS
  FOR EACH ROW
  BEGIN
     DELETE FROM USERS WHERE USERNAME = USERS.USERNAME;
     END;
|
delimiter ;

delimiter |
CREATE TRIGGER NEW_COMIC AFTER INSERT ON MANGA_COMIC
  FOR EACH ROW
  BEGIN
   INSERT INTO PUBLISHER SET COMIC_NAME = NEW.COMIC_NAME ;
       INSERT INTO AUTHOR SET COMIC_NAME = NEW.COMIC_NAME ;
   INSERT INTO THEME_SONG SET COMIC_NAME = NEW.COMIC_NAME ;
   INSERT INTO SHOW_DETAILS SET COMIC_NAME = NEW.COMIC_NAME ;
   END;
|
delimiter ;


CREATE VIEW RATING AS SELECT COMIC_ID, COMIC_NAME, AUTHOR, RATING FROM MANGA_COMIC;
CREATE VIEW CHAR_INFO AS SELECT CHAR_NAME, COMIC_NAME, GENDER, 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 'databaseDEV01'@'localhost' IDENTIFIED BY 'DEV1pass';
CREATE USER 'read_dev1'@'localhost' IDENTIFIED BY 'read_dev1pass';
CREATE USER 'read_dev2'@'localhost' IDENTIFIED BY 'read_dev2pass';

GRANT 'DB_developer' TO 'databaseDEV01'@'localhost';
GRANT 'DB_read' TO 'read_dev1'@'localhost', 'read_dev2'@'localhost';


DROP TABLE SHOW_DETAILS;
DROP TABLE THEME_SONG;
DROP TABLE VIDEOS;
DROP TABLE AUTHOR;
DROP TABLE MANGA_COMIC;
```