

## **Database Report-Anime database**

**Report By: Aniket Agarwal 17317437**

I made a database for anime. Anime is hand-drawn and computer animation originating from or associated with Japan.

The word *anime* is the Japanese term for *animation*, which means all forms of animated media. Outside Japan, *anime* refers specifically to animation from Japan or as a Japanese-disseminated animation style often characterized by colourful graphics, vibrant characters and fantastical themes. The culturally abstract approach to the word's meaning may open up the possibility of anime produced in countries other than Japan. For simplicity, many Westerners strictly view anime as a Japanese animation product. Some scholars suggest defining anime as specifically or quintessentially Japanese may be related to a new form of Orientalism.

The database is modelled relationally as much as possible. I have made as few attributes as I can in each table. The particular attributes and relations can be viewed in the following pages of diagrams.

The Anime table contains the name of the anime, type (if it's a TV series, movie or OVA), number of episodes, rating (of anime), and creator. Each anime name is Unique and cannot be NULL so I defined it as the primary key.

The Character\_list table contains the name of the character, its gender, name of the anime it has appeared in and its rating (of the character).

The Creator\_list table contains the name of the creator, its gender, name of the anime he/she has produced, date of birth, date of death and its rating (of the creator).

The Song\_list table contains the name of the song, its performer, length of the song, name of the anime it has appeared in and its rating (of the song).

The Group\_list table contains the name of the group, its preferred name, name of the anime it has produced, languages it has produced in, date the group was established, and its rating (of the group that produces the anime).

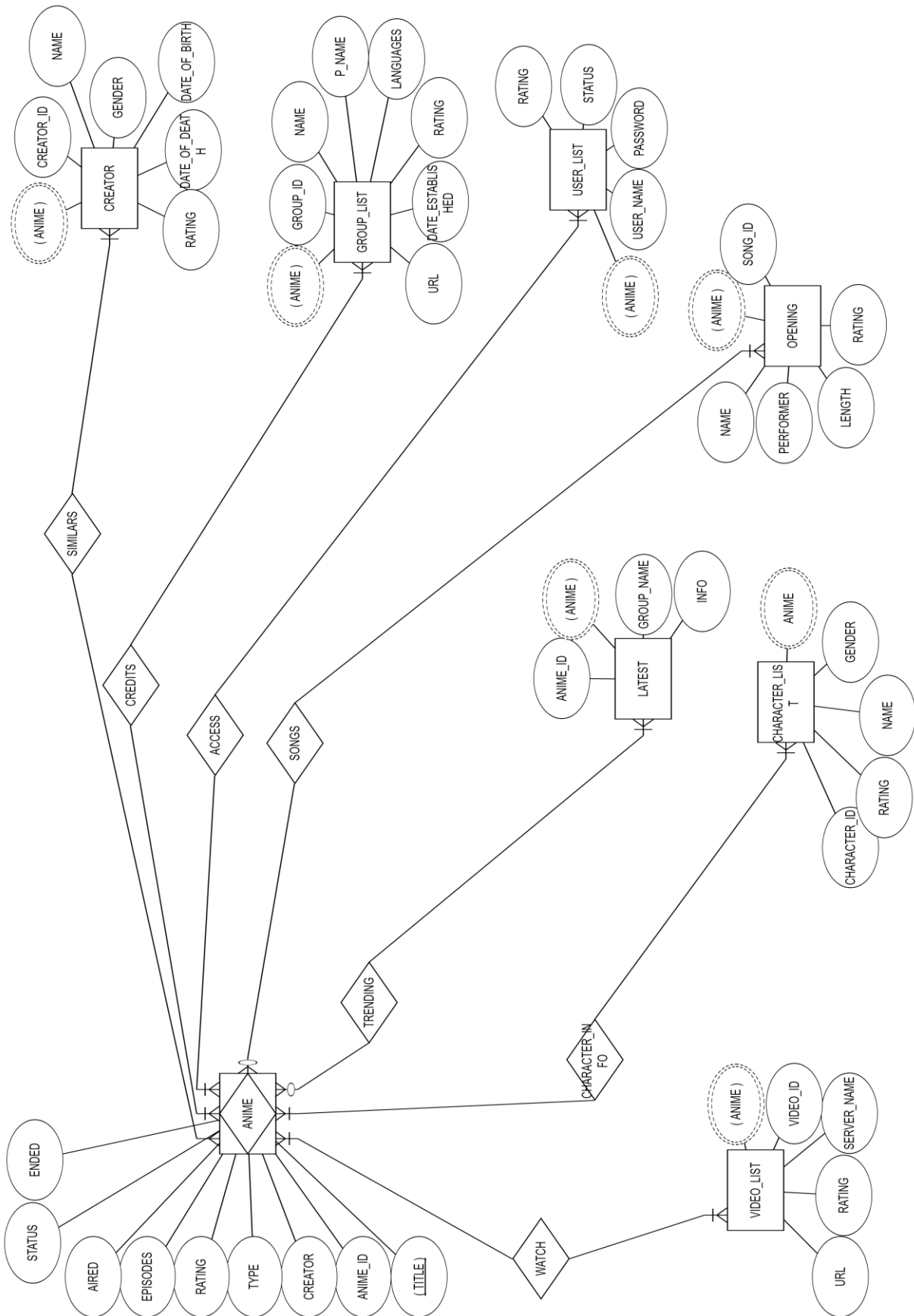
The Latest\_list table contains the name of the anime, the group that produced it, and its information.

The Video\_list table contains the name of the anime, the link to the anime, the server-name associated with it and its rating (of the server which is calculated on the basis of quality of video and availability of server and the number of clients it can hosts).

The Character\_list table contains the name of the character, its gender, name of the anime it has appeared in and its rating (of the character).

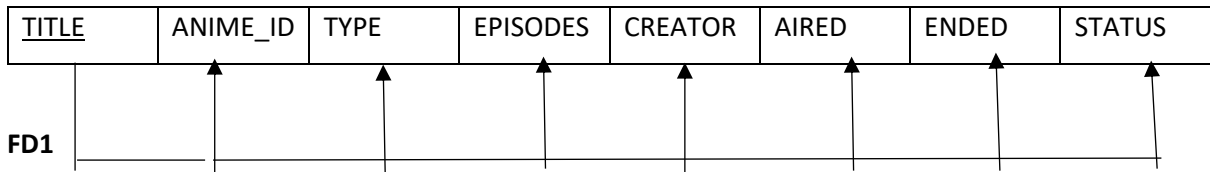
The User\_list table contains the user\_name, its password, name of the anime the user has watched and the status (if the user is still watching it or he has watched it or he will watch it in the future) its rating (of the anime given by that particular user).

# ENTITY-RELATIONSHIP DIAGRAM

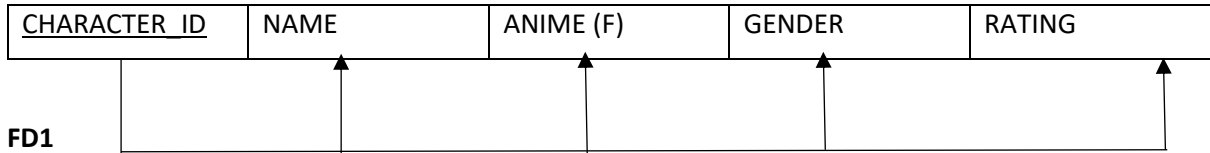


## FUNCTIONAL DEPENDENCY

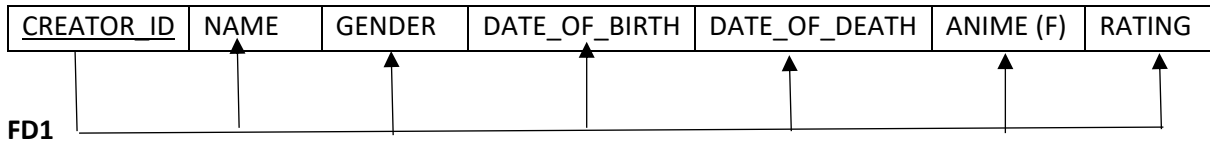
### **ANIME**



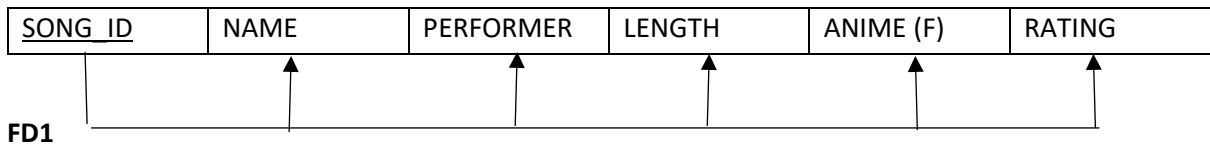
### **CHARACTER\_LIST**



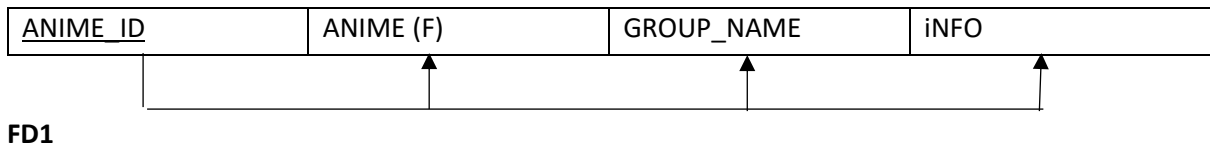
### **CREATOR**



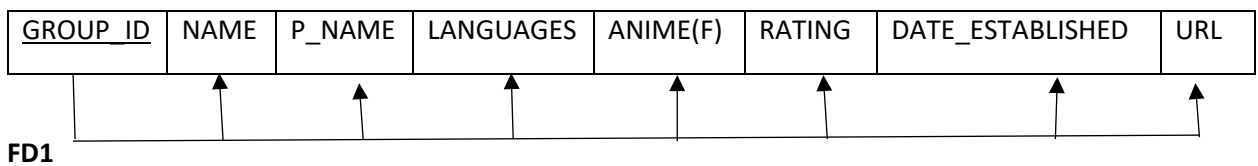
### **OPENING**



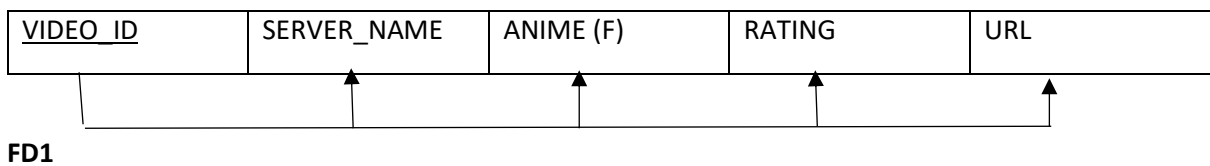
### **LATEST**



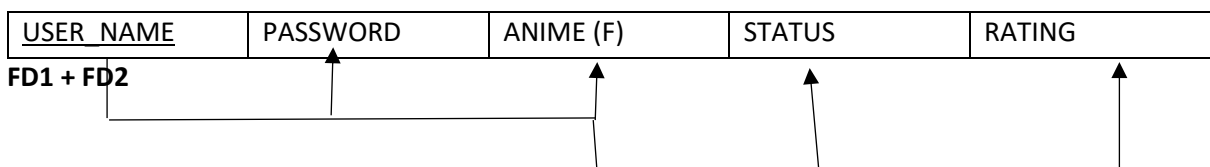
### **GROUP\_LIST**



### **VIDEO\_LIST**



### **USER\_LIST**



## Normalization

The tables I used were generally already normalized. I did add in two tables with composite primary keys to model the many to many relationships and by having them the many to many attributes were normalized.

## Semantic Constraints

The first semantic constraint I set was a constraint on the number that could be entered for a type of an anime, this made sure that the type of anime was either movie, tv series or an OVA.

```
CONSTRAINT CHK_TYPE CHECK ((TYPE = 'OVA') OR (TYPE = 'MOVIE') OR (TYPE = 'TV SERIES'));
```

I also have a semantic constraint on the Rating entity in almost every table. This will make sure that the rating is less than 10.

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

Other constraints that I have put into the database are NOT NULL for most entity entries and of course primary and foreign key constraints. I have also used Check constraint in other tables as well.

## Trigger

The trigger I wrote occurs when an Anime is added and when User is deleted from the User\_list table

```
CREATE TRIGGER addUSER AFTER INSERT ON VIDEO_LIST
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO USER_LIST SET ANIME = NEW.ANIME ;
```

```
INSERT INTO USER_LIST SET USER_NAME = NEW.SERVER_NAME ;
```

```
END;
```

```
CREATE TRIGGER V_USER BEFORE DELETE ON USER_LIST
```

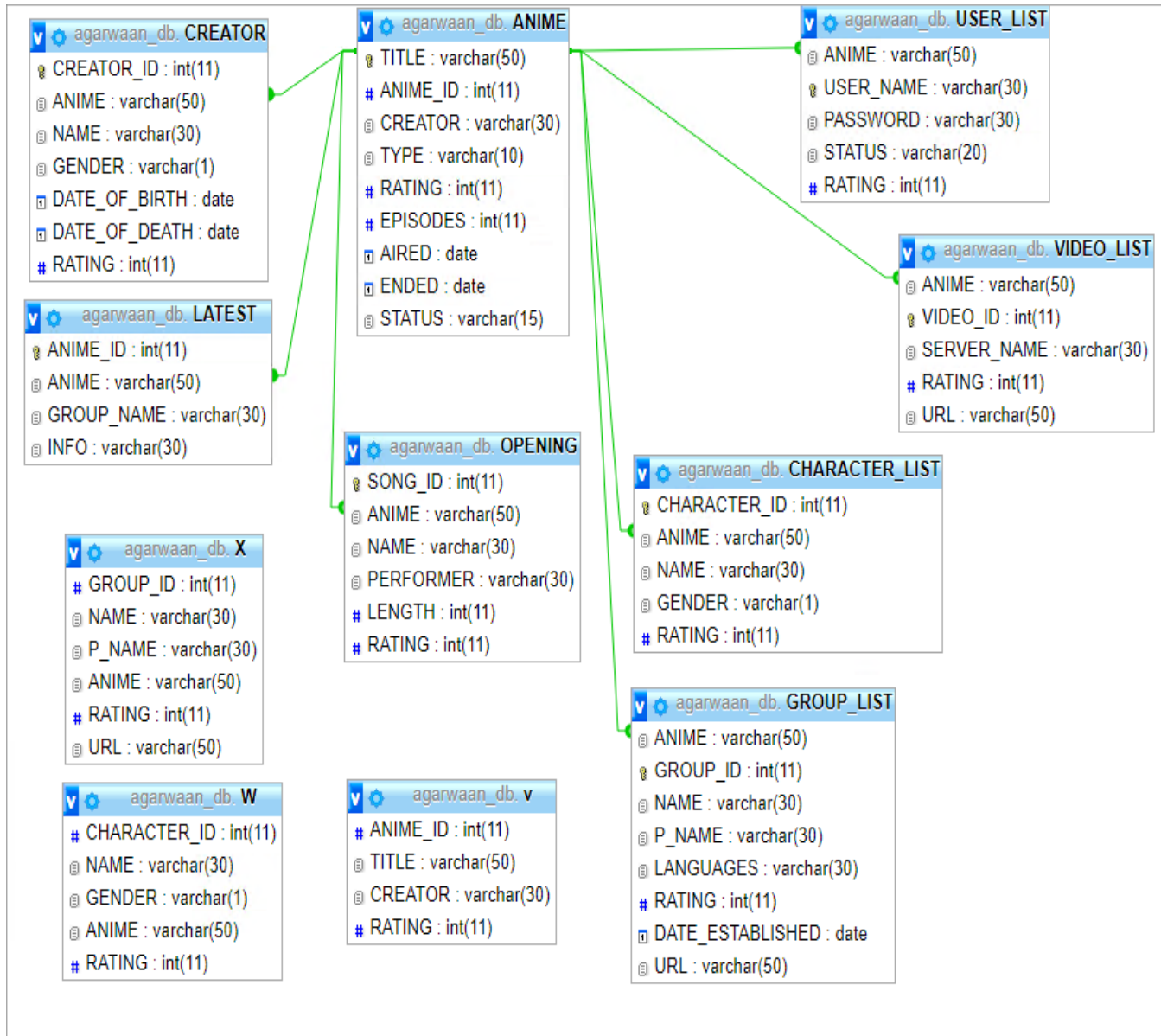
```
FOR EACH ROW
```

```
BEGIN
```

```
DELETE FROM CREATOR WHERE ANIME = USER_LIST.ANIME;
```

```
END;
```

## Relational Schema Diagram



## Security

To add security to my database only Gon and Changer are granted the abilities given with their particular role. For example the changer can only see the view “who makes what where” table which only tells them what their colleagues watch. This hides any sensitive information and gives them the information they need about their fellow Gon. Only the changer is granted the right to create and delete tables. This is because he is trusted with this information. The changer role is identified by the password that is entered for their account. The changer’s password is “killua”.

Create ROLE Changer IDENTIFIED by killua;

Grant create table to Changer;

Grant delete table to Changer;

REVOKE CREATE TABLE FROM Changer;

Grant Changer to Gon;

## Appendix:

--ANIME DATABASE--

--CREATE TABLES--

```
CREATE TABLE ANIME(  
  TITLE VARCHAR(50),  
  ANIME_ID INT,  
  CREATOR VARCHAR(30),  
  TYPE VARCHAR(10),  
  RATING INT,  
  EPISODES INT,  
  AIRED DATE,  
  ENDED DATE,  
  CONSTRAINT ANIME_PK PRIMARY KEY (TITLE),  
  CONSTRAINT CHK_TYPE CHECK ((TYPE = 'OVA') OR (TYPE = 'MOVIE') OR (TYPE = 'TV SERIES')),  
  CONSTRAINT CHK_RATING CHECK (RATING<=10.00)  
);
```

```
ALTER TABLE ANIME  
ADD COLUMN STATUS VARCHAR(15);
```

```
CREATE TABLE CHARACTER_LIST(  
  CHARACTER_ID INT,  
  ANIME VARCHAR(50),  
  NAME VARCHAR(30),  
  GENDER VARCHAR(1),  
  RATING INT,  
  CONSTRAINT CHARACTER_PK PRIMARY KEY (CHARACTER_ID),  
  CONSTRAINT CHK_GENDER CHECK ((GENDER = 'F') OR (GENDER = 'M')),  
  CONSTRAINT CHK_RATING CHECK (RATING<=10.00)  
);
```

```
CREATE TABLE CREATOR(  
  CREATOR_ID INT,  
  ANIME VARCHAR(50),  
  NAME VARCHAR(30),  
  GENDER VARCHAR(1),  
  DATE_OF_BIRTH DATE,  
  DATE_OF_DEATH DATE,  
  RATING INT,  
  CONSTRAINT CREATOR_PK PRIMARY KEY (CREATOR_ID),  
  CONSTRAINT CHK_GENDER CHECK ((GENDER = 'F') OR (GENDER = 'M')),  
  CONSTRAINT CHK_RATING CHECK (RATING<=10.00)  
);
```

```
CREATE TABLE OPENING(  
  SONG_ID INT,  
  ANIME VARCHAR(50),  
  NAME VARCHAR(30),  
  PERFORMER VARCHAR(30),  
  LENGTH INT,
```

```

        RATING INT,
        CONSTRAINT SONG_PK PRIMARY KEY (SONG_ID),
        CONSTRAINT CHK_LENGTH CHECK (LENGTH <= 2.00),
        CONSTRAINT CHK_RATING CHECK (RATING<=10.00)
    );

```

```

CREATE TABLE LATEST(
    ANIME_ID INT,
    ANIME VARCHAR(50),
    GROUP_NAME VARCHAR(30),
    INFO VARCHAR(30),
    CONSTRAINT LATEST_PK PRIMARY KEY (ANIME_ID)
);

```

```

CREATE TABLE GROUP_LIST(
    ANIME VARCHAR(50),
    GROUP_ID INT,
    NAME VARCHAR(30),
    P_NAME VARCHAR(30),
    LANGUAGES VARCHAR(30),
    RATING INT,
    DATE_ESTABLISHED DATE,
    URL VARCHAR(50),
    CONSTRAINT GROUP_PK PRIMARY KEY (GROUP_ID),
    CONSTRAINT CHK_RATING CHECK (RATING<=10.00)
);

```

```

CREATE TABLE VIDEO_LIST(
    ANIME VARCHAR(50),
    VIDEO_ID INT,
    SERVER_NAME VARCHAR(30),
    RATING INT,
    URL VARCHAR(50),
    CONSTRAINT VIDEO_PK PRIMARY KEY (VIDEO_ID),
    CONSTRAINT CHK_RATING CHECK (RATING<=10.00)
);

```

```

CREATE TABLE USER_LIST(
    ANIME VARCHAR(50),
    USER_NAME VARCHAR(30),
    PASSWORD VARCHAR(30),
    STATUS VARCHAR(20),
    RATING INT,
    CONSTRAINT USER_NAME PRIMARY KEY (USER_NAME),
    CONSTRAINT STATUS CHECK ((STATUS='WATCHING') OR (STATUS='WATCHED') OR (STATUS='WILL WATCH')),
    CONSTRAINT CHK_RATING CHECK (RATING<=10.00)
);

```

```

--SELECT--
SELECT * FROM ANIME;

```



```
SELECT * FROM CHARACTER_LIST;
SELECT * FROM CREATOR;
SELECT * FROM GROUP_LIST;
SELECT * FROM LATEST;
SELECT * FROM OPENING;
SELECT * FROM USER_LIST;
SELECT * FROM VIDEO_LIST;
```

--DROP--

```
DROP TABLE CHARACTER_LIST;
DROP TABLE CREATOR;
DROP TABLE GROUP_LIST;
DROP TABLE LATEST;
DROP TABLE OPENING;
DROP TABLE USER_LIST;
DROP TABLE VIDEO_LIST;
DROP TABLE ANIME;
```

--FOREIGN KEYS--

```
ALTER TABLE CHARACTER_LIST
ADD CONSTRAINT ANIME_FK
FOREIGN KEY(ANIME)
REFERENCES ANIME(TITLE);
```

```
ALTER TABLE CREATOR
ADD CONSTRAINT ANIME_FK1
FOREIGN KEY(ANIME)
REFERENCES ANIME(TITLE);
```

```
ALTER TABLE GROUP_LIST
ADD CONSTRAINT ANIME_FK2
FOREIGN KEY(ANIME)
REFERENCES ANIME(TITLE);
```

```
ALTER TABLE LATEST
ADD CONSTRAINT ANIME_FK3
FOREIGN KEY(ANIME)
REFERENCES ANIME(TITLE);
```

```
ALTER TABLE OPENING
ADD CONSTRAINT ANIME_FK4
FOREIGN KEY(ANIME)
REFERENCES ANIME(TITLE);
```

```
ALTER TABLE USER_LIST
ADD CONSTRAINT ANIME_FK5
FOREIGN KEY(ANIME)
REFERENCES ANIME(TITLE);
```

```
ALTER TABLE VIDEO_LIST
ADD CONSTRAINT ANIME_FK6
```

FOREIGN KEY(ANIME)  
REFERENCES ANIME(TITLE);

--INSERTS--

INSERT INTO ANIME VALUES('NARUTO ORIGINAL',1,'Masashi Kishimoto','TV SERIES',9.27,220,'2002-10-03','2007-02-08','COMPLETED');

INSERT INTO ANIME VALUES('NARUTO SHIPPUDEN',2,'Masashi Kishimoto','TV SERIES',9.81,500,'2007-02-15','2017-03-23','COMPLETED');

INSERT INTO ANIME VALUES('BORUTO:NARUTO THE MOVIE',3,'UKYO KODACHI','MOVIE',9.03,4,'2015-08-07','2015-08-07','COMPLETED');

INSERT INTO ANIME VALUES('BORUTO:NARUTO NEXT GENERATIONS',4,'UKYO KODACHI','TV SERIES',8.53,83,'2017-04-05','2018-11-26','ON-GOING');

INSERT INTO ANIME VALUES('The Day Naruto Became Hokage',5,'Masashi Kishimoto','OVA',8.53,1,'2016-05-04','2016-05-04','COMPLETED');

INSERT INTO ANIME VALUES('BOKU NO HERO ACADEMIA',6,'Kohei Horikoshi','TV SERIES',9.53,75,'2016-05-04','','ON-GOING');

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

INSERT INTO CHARACTER\_LIST VALUES( 1, 'NARUTO ORIGINAL', 'NARUTO', 'M', 9.43);

INSERT INTO CHARACTER\_LIST VALUES( 2, 'NARUTO ORIGINAL', 'SASUKE', 'M', 8.37);

INSERT INTO CHARACTER\_LIST VALUES( 3, 'NARUTO ORIGINAL', 'SAKURA', 'F', 8.67);

INSERT INTO CHARACTER\_LIST VALUES( 4, 'NARUTO ORIGINAL', 'KAKASHI', 'M', 8.63);

INSERT INTO CHARACTER\_LIST VALUES( 5, 'NARUTO ORIGINAL', 'YAMATO', 'M', 8.33);

INSERT INTO CREATOR VALUES(1,'NARUTO ORIGINAL','Masashi Kishimoto','M','1974-11-08','',7.56);

INSERT INTO CREATOR VALUES(2,'NARUTO SHIPPUDEN','Masashi Kishimoto','M','1974-11-08','',7.56);

INSERT INTO CREATOR VALUES(3,'BORUTO:NARUTO NEXT GENERATIONS','UKYO KODACHI','M','1979-04-01','',7.56);

INSERT INTO CREATOR VALUES(4,'BOKU NO HERO ACADEMIA','Kohei Horikoshi','M','1987-07-09','',9);

INSERT INTO CREATOR VALUES(5,'KIMI NO NA WA','MAKATO SHINKAI','M','1983-02-19','',10);

INSERT INTO OPENING VALUES(1,'NARUTO ORIGINAL','No Boy,No Cry','Stance Punks',3.33,7.56);

INSERT INTO OPENING VALUES(2,'NARUTO SHIPPUDEN','BLUE BIRD','Ikimono-Gakari',3.36,7.83);

INSERT INTO OPENING VALUES(3,'BORUTO:NARUTO NEXT GENERATIONS','Lonely Go!','Brian the Sun',3.39,7.83);

INSERT INTO OPENING VALUES(4,'NARUTO SHIPPUDEN','DIVER','ELENA',2.39,8);

INSERT INTO OPENING VALUES(5,'KIMI NO NA WA','GO ON','THE TEANGENT',3.37,10);

INSERT INTO LATEST VALUES(1,'BORUTO:NARUTO NEXT GENERATIONS','Tv Tokyo','The ninja adventures continue with Naruto's son, Boruto!Naruto was a young shinobi with an incorrigible knack for mischief. He achieved his dream to become the greatest ninja in his village, and now his face sits atop the Hokage monument. But this is not his story... ..');

INSERT INTO LATEST VALUES(2,'NARUTO ORIGINAL','Tv Tokyo','...');

INSERT INTO LATEST VALUES(3,'NARUTO SHIPPUDEN','Tv Tokyo','...');

INSERT INTO LATEST VALUES(4,'KIMI NO NA WA','VIZ MEDIA','...');

INSERT INTO LATEST VALUES(5,'BOKU NO HERO ACADEMIA','YOUPEI','...');

```

INSERT INTO GROUP_LIST VALUES('BORUTO:NARUTO NEXT GENERATIONS',1,'Tv Tokyo','VIZ
MEDIA','JAPANESE,ENGLISH',6.98,"");
INSERT INTO GROUP_LIST VALUES('NARUTO ORIGINAL',2,'Tv Tokyo','VIZ
MEDIA','JAPANESE,ENGLISH',6,"");
INSERT INTO GROUP_LIST VALUES('NARUTO SHIPPUDEN',3,'Tv Tokyo','VIZ
MEDIA','JAPANESE,ENGLISH',6,"");
INSERT INTO GROUP_LIST VALUES('BOKU NO HERO ACADEMIA',4,'Tv Tokyo','VIZ
MEDIA','JAPANESE,ENGLISH',8,"");
INSERT INTO GROUP_LIST VALUES('KIMI NO NA wA',5,'Tv Tokyo','VIZ
MEDIA','JAPANESE,ENGLISH',9,"");

```

```

INSERT INTO VIDEO_LIST VALUES('BORUTO:NARUTO NEXT
GENERATIONS',1,'RAPIDFIRE',6.98,'WWW.GOGOANIME.TV');
INSERT INTO VIDEO_LIST VALUES('NARUTO ORIGINAL',2,'RAPIDFIRE',7,'WWW.GOGOANIME.TV');
INSERT INTO VIDEO_LIST VALUES('NARUTO SHIPPUDEN',3,'RARE',8,'WWW.GOGOANIME.TV');
INSERT INTO VIDEO_LIST VALUES('BOKU NO HERO ACADEMIA',4,'FIRE',9,'WWW.GOGOANIME.TV');
INSERT INTO VIDEO_LIST VALUES('KIMI NO NA WA',5,'RAP',10,'WWW.GOGOANIME.TV');

```

```

INSERT INTO USER_LIST VALUES('BORUTO:NARUTO NEXT
GENERATIONS','MORDRED_002','GOGOTAH','ONGOING',7.54);
INSERT INTO USER_LIST VALUES('NARUTO SHIPPUDEN','007','GOGOTAH','ONGOING',7);
INSERT INTO USER_LIST VALUES('NARUTO ORIGINAL','003','GOGOTAH','ONGOING',8);
INSERT INTO USER_LIST VALUES('BOKU NO HERO ACADEMIA','004','GOGOTAH','ONGOING',9);
INSERT INTO USER_LIST VALUES('KIMI NO NA WA','005','GOGOTAH','ONGOING',10);

```

--TRIGGERS/VIEWS--

```

delimiter |
CREATE TRIGGER addANIME AFTER INSERT ON ANIME
FOR EACH ROW
BEGIN
    INSERT INTO LATEST SET ANIME = NEW.TITLE ;
END;
|

```

delimiter ;

```

delimiter |

CREATE TRIGGER addUSER AFTER INSERT ON VIDEO_LIST
FOR EACH ROW
BEGIN
    INSERT INTO USER_LIST SET ANIME = NEW.ANIME ;
    INSERT INTO USER_LIST SET USER_NAME = NEW.SERVER_NAME ;

END;
|

```

delimiter ;

```
delimiter |  
CREATE TRIGGER V_USER BEFORE DELETE ON USER_LIST  
FOR EACH ROW  
BEGIN  
    DELETE FROM CREATOR WHERE ANIME = USER_LIST.ANIME;  
END;  
|
```

delimiter ;

```
CREATE VIEW v AS SELECT ANIME_ID, TITLE, CREATOR, RATING FROM ANIME;  
CREATE VIEW W AS SELECT CHARACTER_ID, NAME, GENDER, ANIME, RATING FROM  
CHARACTER_LIST;  
CREATE VIEW X AS SELECT GROUP_ID, NAME, P_NAME, ANIME, RATING, URL FROM GROUP_LIST;
```

```
Create ROLE Changer IDENTIFIED by killua;  
Grant create table to Changer;  
Grant delete table to Changer;  
REVOKE CREATE TABLE FROM Changer;
```

```
Grant Changer to Gon;
```