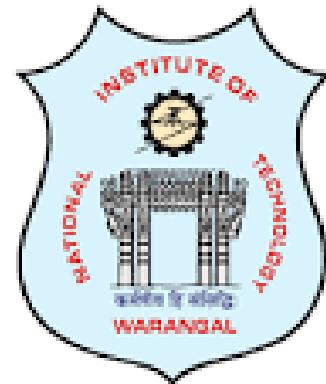


DBMS PROJECT

Team :

Rutvik Pradyumna.K
Sai Lokesh.P

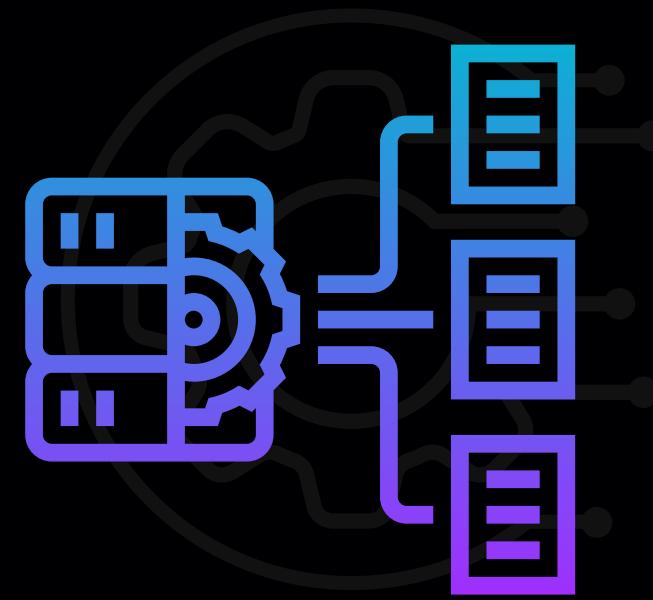
21CSB0B30
21CSB0B40



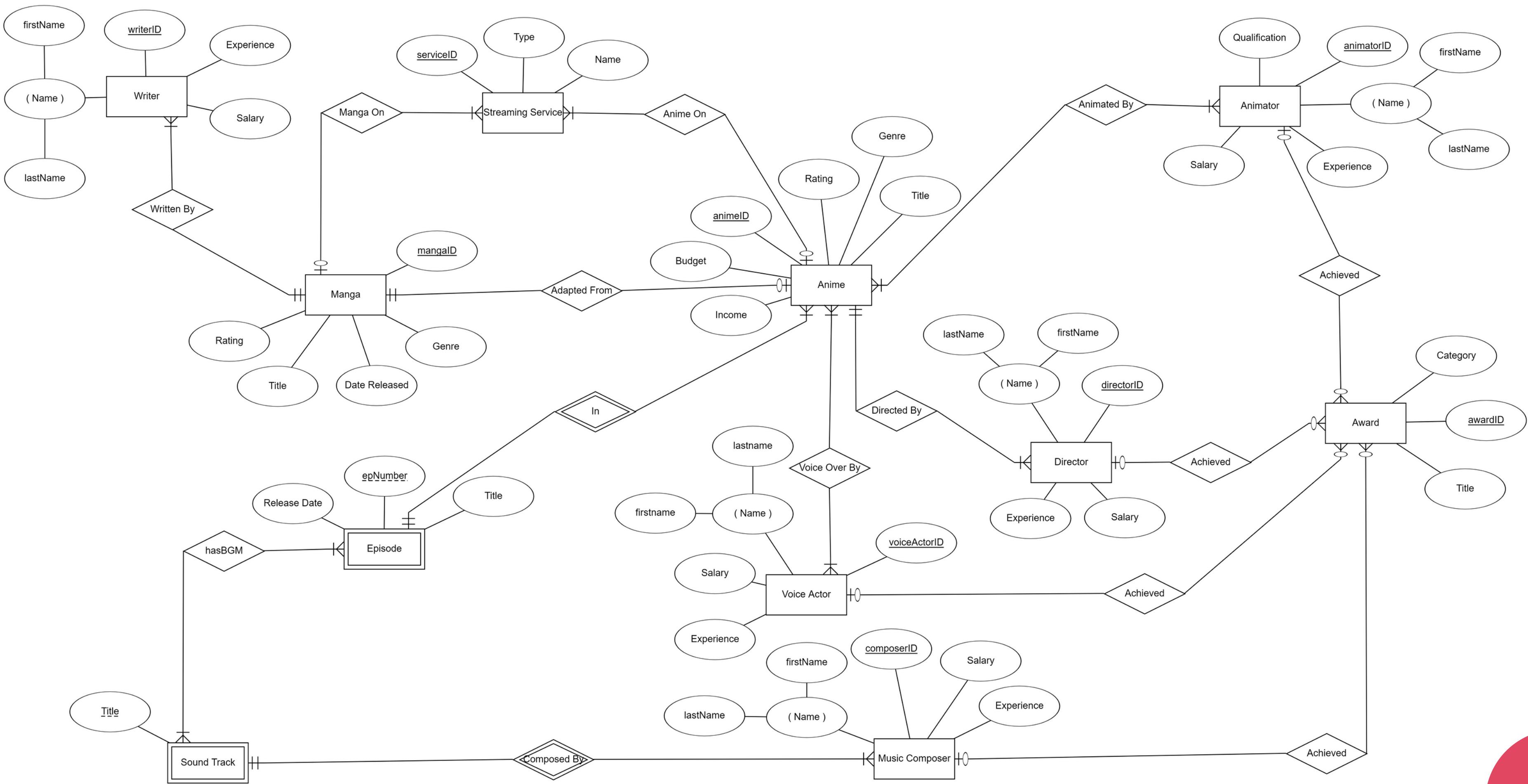
PROBLEM STATEMENT :

Anime Studios...

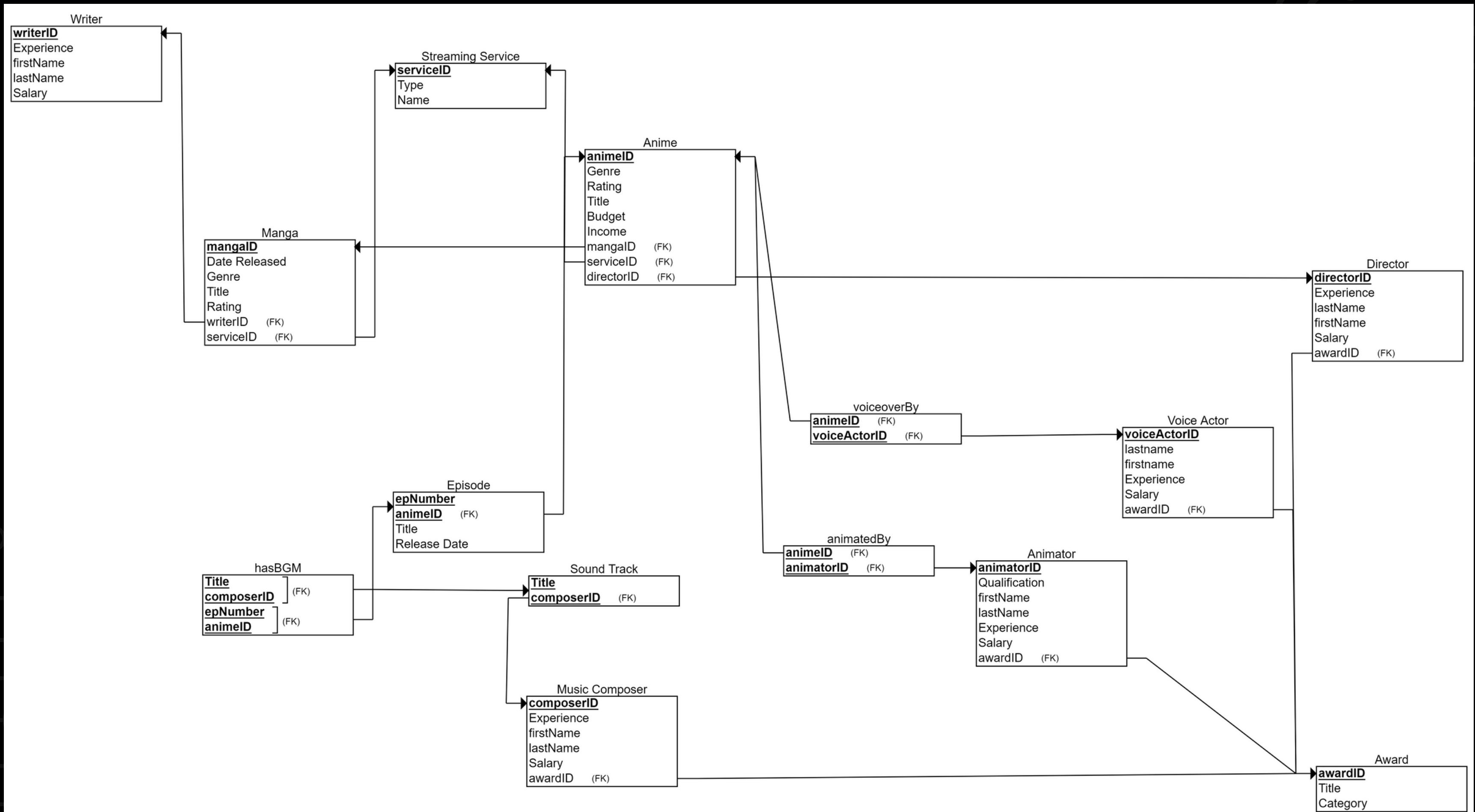
This project involves the design of a database that will store and manage information related to an anime studio. The database will contain crucial and relevant information about writers, directors, manga, anime, awards, soundtracks, release dates, voice actors, genres, and more. The aim of the system is to enable easy access to information for users who are interested in the anime produced by the studio. Additionally, the system will allow for efficient management of the data, ensuring that information is organized, up-to-date, and easily searchable.

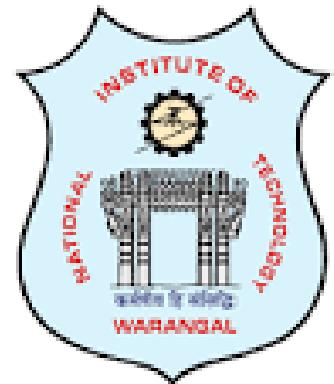


Entity Relationship Model :



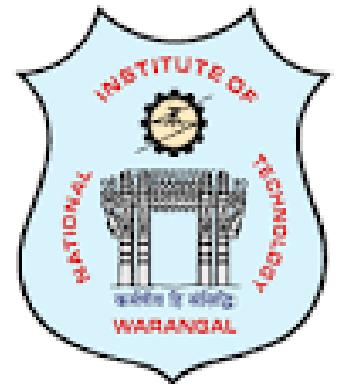
Relational Schema :





Assumptions :

- A writer can write any number of manga's but one manga can be written by only one writer.
- Each anime or manga will be streaming on single streaming platform but not reverse.
- Only few mangas will be adapted and animated as animes and every anime will be from a manga
- A single anime can have many animators and voice-actors but only one director.
- Each director/animator/voice actor/music composer can get award once only but one award can be given to many
- Each episode can have many soundtracks and each soundtrack can be in more than one episodes.
- Each soundtrack will be created by one composer but one composer can create many soundtracks.



Normalisation :

1.streamingService :

serviceID -> {type,name}

2.Writer:

writerID -> {Experience,lastName,firstName,Salary}

3.Director :

directorID -> {Experience,lastName,firstName,Salary,awardID}

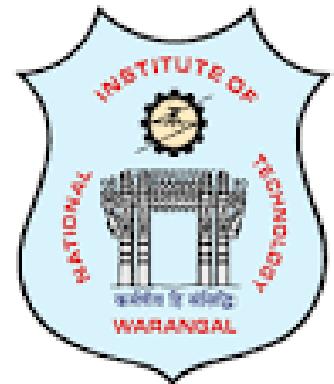
4.Animator :

animID -> {Experience,lastName,firstName,Salary,awardID,qualification}

5.musicComposer :

musicComposerID -> {Experience,lastName,firstName,Salary,awardID,qualification}

-> As these tables are having only one FD they all are in BCNF.



Normalisation :

6.voiceActor :

voiceActorID -> {Experience,lastName,firstName,Salary,awardID}

7.Award :

awardID -> {title,category}

8.voiceOverBy

9.animatedBy

10.soundTrack

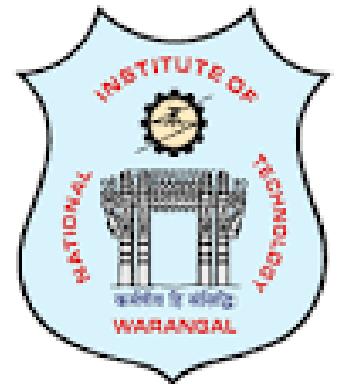
11.hasBGM :

{title,composerID,epNumber,animeID} -> {title,composerID,epNumber,animeID}

12.Manga :

mangaID -> {dateReleased,genre,title,rating,writerID,serviceID}

-> Here, some tables are having trivial and single FD's others are binary so they will be in BCNF.



Normalisation :

13.Anime :

$\text{animID} \rightarrow \{\text{genre}, \text{rating}, \text{title}, \text{budget}, \text{income}, \text{mangaID}, \text{serviceID}, \text{directorID}\}$

$\text{mangaID} \rightarrow \{\text{genre}, \text{rating}, \text{title}, \text{budget}, \text{income}, \text{animID}, \text{serviceID}, \text{directorID}\}$

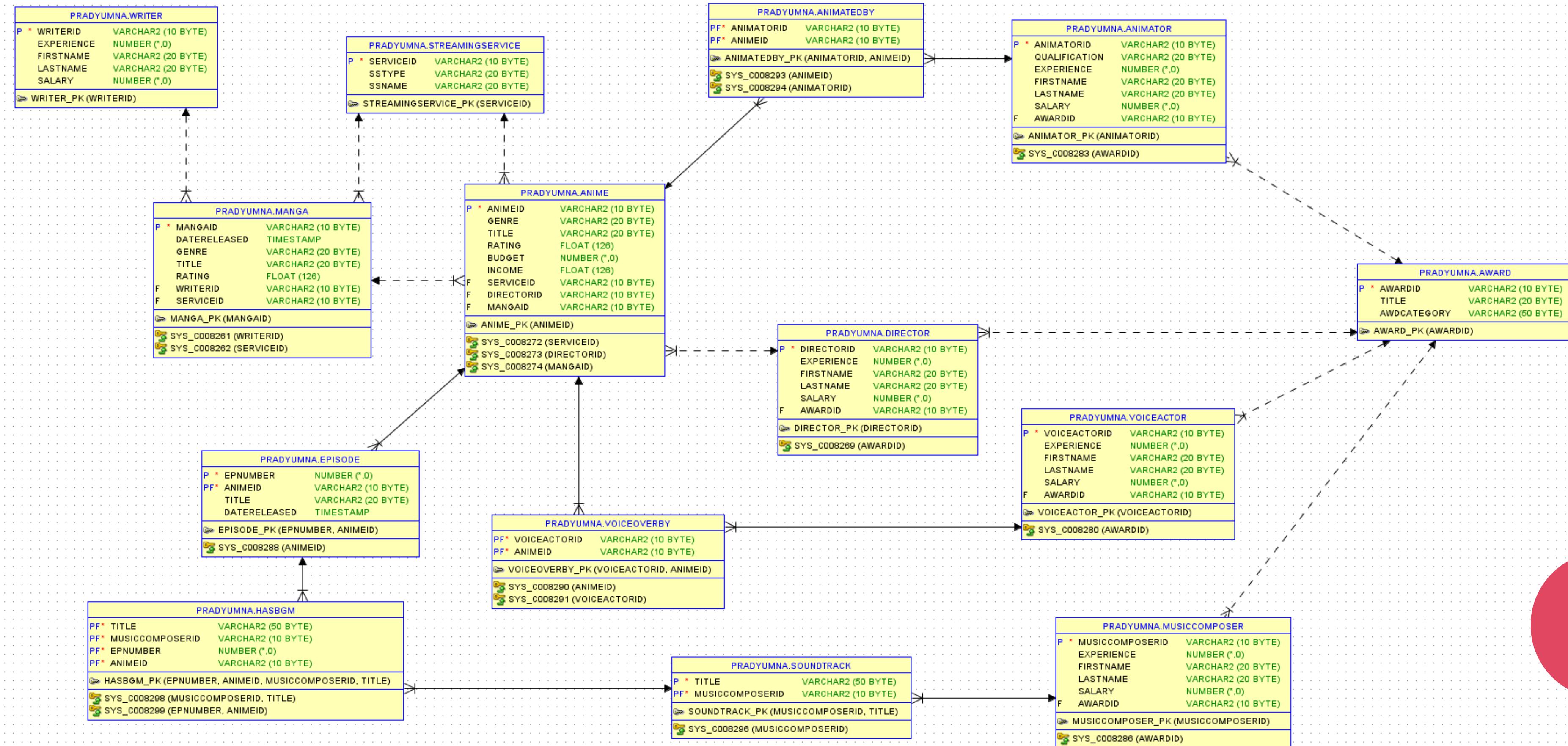
-> LHS of FD's are super keys so, it is in BCNF

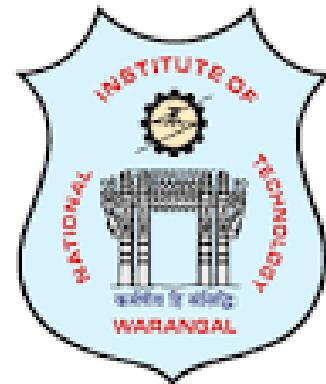
14.episode :

$\{\text{epNumber}, \text{animID}\} \rightarrow \{\text{title}, \text{releaseDate}\}$

-> As this table is having only one FD they all are in BCNF.

Relational Schema :





Tables :

1) streamingService :

Name	Null?	Type
SERVICEID	NOT NULL	VARCHAR2 (10)
SSTYPE		VARCHAR2 (20)
SSNAME		VARCHAR2 (20)

2) Award :

Name	Null?	Type
AWARDID	NOT NULL	VARCHAR2 (10)
TITLE		VARCHAR2 (50)
AWDCATEGORY		VARCHAR2 (50)

3) Writer :

Name	Null?	Type
WRITERID	NOT NULL	VARCHAR2 (10)
EXPERIENCE		NUMBER (38)
FIRSTNAME		VARCHAR2 (20)
LASTNAME		VARCHAR2 (20)
SALARY		NUMBER (38)

4) Manga :

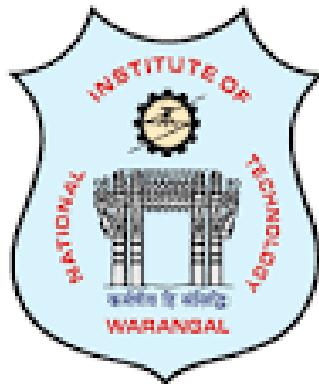
Name	Null?	Type
MANGAID	NOT NULL	VARCHAR2 (10)
DATERELEASED		TIMESTAMP (6)
GENRE		VARCHAR2 (20)
TITLE		VARCHAR2 (20)
RATING		FLOAT (126)
WRITERID		VARCHAR2 (10)
SERVICEID		VARCHAR2 (10)

5) Director :

Name	Null?	Type
DIRECTORID	NOT NULL	VARCHAR2 (10)
EXPERIENCE		NUMBER (38)
FIRSTNAME		VARCHAR2 (20)
LASTNAME		VARCHAR2 (20)
SALARY		NUMBER (38)
AWARDID		VARCHAR2 (10)

6) voiceActor :

Name	Null?	Type
VOICEACTORID	NOT NULL	VARCHAR2 (10)
EXPERIENCE		NUMBER (38)
FIRSTNAME		VARCHAR2 (20)
LASTNAME		VARCHAR2 (20)
SALARY		NUMBER (38)
AWARDID		VARCHAR2 (10)



Tables :

7) Animator :

Name	Null?	Type
ANIMATORID	NOT NULL	VARCHAR2 (10)
QUALIFICATION		VARCHAR2 (80)
EXPERIENCE		NUMBER (38)
FIRSTNAME		VARCHAR2 (20)
LASTNAME		VARCHAR2 (20)
SALARY		NUMBER (38)
AWARDID		VARCHAR2 (10)

8) musicComposer :

Name	Null?	Type
MUSICCOMPOSERID	NOT NULL	VARCHAR2 (10)
EXPERIENCE		NUMBER (38)
FIRSTNAME		VARCHAR2 (20)
LASTNAME		VARCHAR2 (20)
SALARY		NUMBER (38)
AWARDID		VARCHAR2 (10)

9) voiceOverBy :

Name	Null?	Type
VOICEACTORID	NOT NULL	VARCHAR2 (10)
ANIMEID	NOT NULL	VARCHAR2 (10)

10) animatedBy :

Name	Null?	Type
ANIMATORID	NOT NULL	VARCHAR2 (10)
ANIMEID	NOT NULL	VARCHAR2 (10)

11) soundTrack :

Name	Null?	Type
TITLE	NOT NULL	VARCHAR2 (80)
MUSICCOMPOSERID	NOT NULL	VARCHAR2 (10)

12) hasBGM :

Name	Null?	Type
TITLE	NOT NULL	VARCHAR2 (80)
MUSICCOMPOSERID	NOT NULL	VARCHAR2 (10)
EPNUMBER	NOT NULL	NUMBER (38)
ANIMEID	NOT NULL	VARCHAR2 (10)



Tables :

13) Anime :

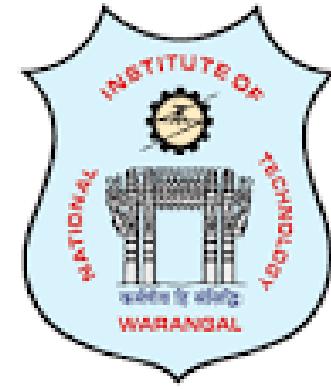
Name	Null?	Type
ANIMEID	NOT NULL	VARCHAR2 (10)
GENRE		VARCHAR2 (20)
TITLE		VARCHAR2 (20)
RATING		FLOAT (126)
BUDGET		NUMBER (38)
INCOME		NUMBER (38)
SERVICEID		VARCHAR2 (10)
DIRECTORID		VARCHAR2 (10)
MANGAID		VARCHAR2 (10)

14) Episode :

Name	Null?	Type
EPNUMBER	NOT NULL	NUMBER (38)
ANIMEID	NOT NULL	VARCHAR2 (10)
TITLE		VARCHAR2 (80)
DATERELEASED		TIMESTAMP (6)

Triggers Used :

- > We used a trigger namely 'checkRating' to restrict the type of value and range of rating of Anime Table. Here the rating should be of type 'x.x' and it should be greater than or equal to 0.1 and less than or equal to 9.9.
- > So, we used user defined exceptions i.e, boundExpRating is raised when the value of rating is not in range of 0.1 and 9.9 and made as 0.1 if it is less than 0.1 and made as 9.9 if it is greater than 9.9.
- > inAppExpRating is another user defined exception we used which is raised when rating is not in type 'x.x' and it is rounded to one decimal place.



Trigger Code :



```
/  
SET SERVEROUTPUT ON;  
CREATE OR REPLACE TRIGGER checkRating  
BEFORE INSERT ON anime  
FOR EACH ROW  
DECLARE  
boundExpRating EXCEPTION;  
inAppExpRating EXCEPTION;  
dupRating FLOAT;  
BEGIN  
IF :new.rating > 9.9 OR :new.rating < 0.1 THEN  
    RAISE boundExpRating;  
END IF;  
dupRating := :new.rating;  
IF dupRating != dupRating*100 THEN  
    RAISE inAppExpRating;  
END IF;  
  
EXCEPTION  
WHEN boundExpRating THEN  
IF :new.rating < 0.1 THEN  
:new.rating := 0.1;  
ELSIF :new.rating > 9.9 THEN  
:new.rating := 9.9;  
END IF;  
WHEN inAppExpRating THEN  
:new.rating := ROUND(:new.rating,1);  
END;  
/
```



Queries :

- 1) SQL Query to show all the episodes and their corresponding BGM's from 'attack on titan' anime and order them by episode number.

```
SELECT hasBGM.title AS bgmTitle,epNumber  
FROM anime JOIN hasBGM  
ON anime.animelD=hasBGM.animelD  
WHERE anime.title LIKE 'attack on titan'  
ORDER BY epNumber;
```

BGMTITLE	EPNUMBER
1 Vogel im Käfig	1
2 Vogel im Käfig	2
3 Guren no Yumiya	3
4 Ashes on The Fire	4
5 Guren no Yumiya	5



Queries :

2) SQL Query to show all the mangas that are not animated and their writer.

```
SELECT mangaID,mangaTitle,writer.writerID,  
writer.firstName,writer.lastName FROM  
(SELECT manga.mangaID,manga.title AS  
mangaTitle,writerID FROM  
manga LEFT JOIN anime  
ON manga.mangaID=anime.mangaID  
WHERE anime.mangaID IS NULL) T JOIN writer  
ON writer.writerID=T.writerID;
```

MANGAID	MANGATITLE	WRITERID	FIRSTNAME	LASTNAME
1 mg007	DR. SLUMP	wtr001	Eiichiro	Oda
2 mg008	Slam Dunk	wtr008	Naoki	Urasawa
3 mg009	The Heart of Thomas	wtr010	Moto	Hagio
4 mg010	SOLO LEVELING	wtr009	Jang	Sung-Rak

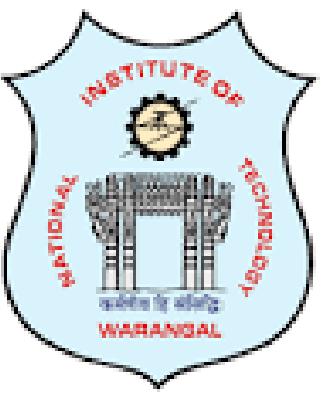


Queries :

3) SQL Querie to find awards recieived by animators of 'NARUTO'

```
SELECT animatorID,firstName,lastName,  
award.awardID,title FROM  
(SELECT animator.animatorID,  
firstName,lastName,awardID FROM  
(SELECT * FROM  
anime JOIN animatedBy  
ON anime.animelD=animatedBy.animelD  
WHERE title LIKE 'NARUTO') T1 JOIN animator  
ON T1.animatorID=animator.animatorID  
WHERE awardID IS NOT NULL) T2 JOIN award  
ON T2.awardID=award.awardID;
```

	ANIMATORID	FIRSTNAME	LASTNAME	AWARDID	TITLE
1	amt005	Junichi	Hayama	awd001	MetaVerse Achievement Award
2	amt007	Tomonaga	KeKazuhide	awd007	Imagination Architect Prize



THANK YOU

