**FOR ERD Diagram**

movies

--

movie\_id PK int

title varchar(100)

revenue bigint

tagline varchar(300)

average\_vote decimal(5)

popularity decimal(9)

release\_date date

actor

--

actor\_id PK int FK -< credits\_actor.actor\_id

actor varchar(50)

credits\_actor

--

movie\_id PK int FK >- movies.movie\_id

actor\_id PK int

directors

--

director\_id pk int FK -< movieid\_director\_id.director\_id

director varchar(100)

movieid\_director\_id

--

movie\_id PK int FK >- movies.movie\_id

director\_id PK int

emoji\_genre

--

genre\_id PK int FK -< movieid\_genre\_ids.genre\_id

genre varchar(100)

alias varchar(100)

genre\_emoji varchar(100)

movieid\_genre\_ids

--

movie\_id PK int FK >- movies.movie\_id

genre\_id PK int

keywords

--

keyword\_id PK int FK -< movieids\_kw.keyword\_id

keyword varchar(50)

movieids\_kw

--

movie\_id PK int FK >- movies.movie\_id

keyword\_id PK int

**SQL DATABASE**

-- Exported from QuickDBD: https://www.quickdatabasediagrams.com/

-- NOTE! If you have used non-SQL datatypes in your design, you will have to change these here.

CREATE TABLE "movies" (

"movie\_id" int NOT NULL,

"title" varchar(100) NOT NULL,

"revenue" bigint NOT NULL,

"tagline" varchar(300) NOT NULL,

"average\_vote" decimal(5) NOT NULL,

"popularity" decimal(9) NOT NULL,

"release\_date" date NOT NULL,

CONSTRAINT "pk\_movies" PRIMARY KEY (

"movie\_id"

)

);

select \* FROM movies;

select distinct movie\_id, count(movie\_id) as "Movie ID count"

FROM movies

group by movie\_id

order by movie\_id;

select movie\_id

FROM movies

where movie\_id = 315011;

-----------------------------------------------------------------------------------------

CREATE TABLE "credits\_actor" (

"movie\_id" int NOT NULL,

"actor\_id" int NOT NULL

);

select \* FROM credits\_actor;

ALTER TABLE credits\_actor ADD PRIMARY KEY (movie\_id, actor\_id); --ERROR: Key (movie\_id, actor\_id)=(67373, 3085) is duplicated.could not create unique index "credits\_actor\_pkey"

select distinct movie\_id, count(movie\_id) as "credits\_actor ID count"

FROM credits\_actor

group by movie\_id

order by movie\_id;

-----------------------------------------------------------------------------------------

CREATE TABLE "actor" (

"actor\_id" int NOT NULL,

"actor" varchar(50) NOT NULL,

CONSTRAINT "pk\_actor" PRIMARY KEY (

"actor\_id"

)

);

select \* FROM actor;

-----------------------------------------------------------------------------------------

CREATE TABLE "movieid\_director\_id" (

"movie\_id" int NOT NULL,

"director\_id" int NOT NULL

);

select \* FROM movieid\_director\_id;

ALTER TABLE movieid\_director\_id ADD PRIMARY KEY (movie\_id, director\_id);

select distinct movie\_id, count(movie\_id) as "movieid\_director\_id ID count"

FROM movieid\_director\_id

group by movie\_id

order by movie\_id;

select movie\_id --Need to delete these

FROM movieid\_director\_id

where movie\_id = 315011;

delete from movieid\_director\_id

where movie\_id = ;

-----------------------------------------------------------------------------------------

CREATE TABLE "directors" (

"director\_id" int NOT NULL,

"director" varchar(100) NOT NULL,

CONSTRAINT "pk\_directors" PRIMARY KEY (

"director\_id"

)

);

select \* FROM directors;

-----------------------------------------------------------------------------------------

CREATE TABLE "movieid\_genre\_ids" (

"movie\_id" int NOT NULL,

"genre\_id" int NOT NULL

);

select \* FROM movieid\_genre\_ids;

ALTER TABLE movieid\_genre\_ids ADD PRIMARY KEY (movie\_id, genre\_id);

-----------------------------------------------------------------------------------------

CREATE TABLE "emoji\_genre" (

"genre\_id" int NOT NULL,

"genre" varchar(100) NOT NULL,

"alias" varchar(100) NOT NULL,

"genre\_emoji" varchar(100) NOT NULL,

CONSTRAINT "pk\_emoji\_genre" PRIMARY KEY (

"genre\_id"

)

);

select \* FROM emoji\_genre;

-----------------------------------------------------------------------------------------

CREATE TABLE "movieids\_kw" (

"movie\_id" int NOT NULL,

"keyword\_id" int NOT NULL

);

select \* FROM movieids\_kw;

ALTER TABLE movieids\_kw ADD PRIMARY KEY (movie\_id, keyword\_id);

-----------------------------------------------------------------------------------------

CREATE TABLE "keywords" (

"keyword\_id" int NOT NULL,

"keyword" varchar(50) NOT NULL,

CONSTRAINT "pk\_keywords" PRIMARY KEY (

"keyword\_id"

)

);

select \* FROM keywords;

-----------------------------------------------------------------------------------------

ALTER TABLE "credits\_actor" ADD CONSTRAINT "fk\_credits\_actor\_movie\_id" FOREIGN KEY("movie\_id")

REFERENCES "movies" ("movie\_id"); --ERROR: Key (movie\_id)=(315011) is not present in table "movies".insert or update on table "credits\_actor" violates foreign key constraint "fk\_credits\_actor\_movie\_id"

-----------------------------------------------------------------------------------------

ALTER TABLE "actor" ADD CONSTRAINT "fk\_actor\_actor\_id" FOREIGN KEY("actor\_id")

REFERENCES "credits\_actor" ("actor\_id"); --ERROR: there is no unique constraint matching given keys for referenced table "credits\_actor"

select distinct actor\_id, count(actor\_id) as "credits\_actor ID count" -- 9394 for both

FROM credits\_actor

group by actor\_id

order by actor\_id;

select distinct actor\_id, count(actor\_id) as "actor ID count"

FROM actor

group by actor\_id

order by actor\_id;

-----------------------------------------------------------------------------------------

ALTER TABLE "movieid\_director\_id" ADD CONSTRAINT "fk\_movieid\_director\_id\_movie\_id" FOREIGN KEY("movie\_id")

REFERENCES "movies" ("movie\_id"); --ERROR: Key (movie\_id)=(315011) is not present in table "movies".insert or update on table "movieid\_director\_id" violates foreign key constraint "fk\_movieid\_director\_id\_movie\_id"

-----------------------------------------------------------------------------------------

ALTER TABLE "directors" ADD CONSTRAINT "fk\_directors\_director\_id" FOREIGN KEY("director\_id")

REFERENCES "movieid\_director\_id" ("director\_id");

select \* FROM movieid\_director\_id;

-----------------------------------------------------------------------------------------

ALTER TABLE "movieids\_kw" ADD CONSTRAINT "fk\_movieids\_kw\_movie\_id" FOREIGN KEY("movie\_id")

REFERENCES "movies" ("movie\_id"); --WORKING

select \* FROM movieids\_kw;

-----------------------------------------------------------------------------------------

ALTER TABLE "keywords" ADD CONSTRAINT "fk\_keywords\_keyword\_id" FOREIGN KEY("keyword\_id")

REFERENCES "movieids\_kw" ("keyword\_id"); --ERROR: there is no unique constraint matching given keys for referenced table "movieids\_kw"

select distinct keyword\_id, count(keyword\_id) as "movieids\_kw ID count" --both coming up 9507

FROM movieids\_kw

group by keyword\_id

order by keyword\_id;

select distinct keyword\_id, count(keyword\_id) as "keywords ID count"

FROM keywords

group by keyword\_id

order by keyword\_id;

----

select keyword\_id, count(\*) as occurance

FROM movieids\_kw

Group by keyword\_id

having count(\*) > 1

order by keyword\_id;

select keyword\_id, count(\*) as occurance

FROM keywords

Group by keyword\_id

having count(\*) > 1

order by keyword\_id;

-----------------------------------------------------------------------------------------

ALTER TABLE "movieid\_genre\_ids" ADD CONSTRAINT "fk\_movieid\_genre\_ids\_movie\_id" FOREIGN KEY("movie\_id")

REFERENCES "movies" ("movie\_id"); --WORKING

select \* FROM movieid\_genre\_ids;

-----------------------------------------------------------------------------------------

ALTER TABLE "emoji\_genre" ADD CONSTRAINT "fk\_emoji\_genre\_genre\_id" FOREIGN KEY("genre\_id")

REFERENCES "movieid\_genre\_ids" ("genre\_id"); --ERROR: there is no unique constraint matching given keys for referenced table "movieid\_genre\_ids"

select distinct genre\_id, count(genre\_id) as "movieid\_genre\_ids ID count" --both coming up 9507

FROM movieid\_genre\_ids

group by genre\_id

order by genre\_id;

select distinct genre\_id, count(genre\_id) as "emoji\_genre ID count"

FROM emoji\_genre

group by genre\_id

order by genre\_id;

----

select genre\_id, count(\*) as occurance

FROM movieid\_genre\_ids

Group by genre\_id

having count(\*) > 1

order by genre\_id;

select genre\_id, count(\*) as occurance

FROM emoji\_genre

Group by genre\_id

having count(\*) > 1

order by genre\_id;

-----------------------------------------------------------------------------------------

select \* FROM movies;

select \* FROM credits\_actor;

select \* FROM actor;

select \* FROM movieid\_director\_id;

select \* FROM directors;

select \* FROM movieid\_genre\_ids;

select \* FROM emoji\_genre;

select \* FROM movieids\_kw;

select \* FROM keywords;ALTER TABLE table\_name ADD PRIMARY KEY (a, b);