Case Study: Movie database

Rahul Khanduri

SQL Essentials

30-07-2023

**Create a new database called “Movie\_case\_study”**

create database movie\_case\_study;

**To work inside the movie database “use command”**

use movie\_case\_study;

**Create “actor” table and insert values accordingly**

create table actor(

Actor\_ID int primary key,

Actor\_F\_name varchar(20),

Actor\_L\_name varchar(20),

Actor\_gender enum('M', 'F')

);

desc actor;

insert into actor values

(101, 'Ryan', 'Gosling', 'M'),

(102, 'Margot', 'Robbie', 'F'),

(103, 'Cillian', 'Murphy', 'M'),

(104, 'Emily', 'Blunt', 'F');

select \* from actor;

**Create “director” table and insert values accordingly**

create table director(

Director\_ID int primary key,

Director\_F\_name varchar(20),

Director\_L\_name varchar(20)

);

desc director;

insert into director values

(201, 'Cristopher', 'Nolan'),

(202, 'Sam', 'Mendes'),

(203, 'Greta', 'Gerwig'),

(204, 'Zoya', 'Akhtar');

select \* from director;

**Create “movie” table and insert values accordingly**

create table movie(

Movie\_ID int primary key,

Title varchar(50),

Year int,

Runtime int,

Language varchar(25),

Release\_date date,

Country\_release varchar(10)

);

desc movie;

insert into movie values

(301, 'Oppenheimer', 2023, 180, 'English', "2023-07-21", 'USA'),

(302, 'Barbie', 2023, 114, 'English', "2023-07-21", 'USA'),

(303, 'Pathaan', 2023, 146, 'Hindi', "2023-01-25", 'India'),

(304, 'RRR', 2022, 182, 'Telugu', "2022-03-25", 'India');

select \* from movie;

**Create “movie\_direction” table and insert values accordingly**

create table movie\_direction(

Director\_ID int, foreign key(Director\_ID) references director(Director\_ID), Movie\_ID int, foreign key(Movie\_ID) references movie(Movie\_ID)

);

desc movie\_direction;

insert into movie\_direction values

(201, 301),

(202, 302),

(203, 303),

(204, 304);

select \* from movie\_direction;

**Create “movie\_cast” table and insert values accordingly**

create table movie\_cast(

Actor\_ID int, foreign key(Actor\_ID) references actor(Actor\_ID),

Movie\_ID int, foreign key(Movie\_ID) references movie(Movie\_ID),

Role varchar(25)

);

desc movie\_cast;

insert into movie\_cast values

(101, 301, 'Main Lead'),

(102, 302, 'Second Lead'),

(103, 303, 'Cameo'),

(104, 304, 'Guest\_Apperance');

select \* from movie\_cast;

**Create “genres” table and insert values accordingly**

create table genres(

Genre\_ID int primary key,

Genres varchar(20)

);

desc genres;

insert into genres values

(401, 'Drama'),

(402, 'Rom-Com'),

(403, 'Action'),

(404, 'Horror');

select \* from genres;

**Create “movie\_genres” table and insert values accordingly**

create table movie\_genres(

Movie\_ID int, foreign key(Movie\_ID) references movie(Movie\_ID),

Genre\_ID int, foreign key(Genre\_ID) references genres(Genre\_ID)

);

desc movie\_genres;

insert into movie\_genres values

(301, 401),

(302, 402),

(303, 403),

(304, 404);

select \* from movie\_genres;

**Create “reviewer” table and insert values accordingly**

create table reviewer(

Review\_ID int primary key,

Review\_name varchar(25)

);

desc reviewer;

insert into reviewer values

(501, 'Anmol Jamwal'),

(502, 'Shan Parasher'),

(503, 'Chris Tuchman'),

(504, 'Jeremy Jahns');

select \* from reviewer;

**Create “rating” table and insert values accordingly**

create table rating(

Movie\_ID int, foreign key(Movie\_ID) references movie(Movie\_ID),

Review\_ID int, foreign key(Review\_ID) references reviewer(Review\_ID), Star\_review enum('1', '2', '3', '4', '5'),

Num\_ratings int

);

insert into rating values

(301, 501, '4', 7),

(302, 502, '3', 6),

(303, 503, '5', 8),

(304, 504, '4', 8);

select \* from rating;