United International University
Department of Computer Science and Engineering

January 1, 2018

Project Report:
Database Management System Laboratory
Fall - 2017

Project Name:
IMDB-UIU

Team ID:

Project Members:

SL	***ID***	***Section***	***Name***
1	011151289	C	Safi Ul Sahid
2			

1 Project Description

IMDB-UIU is a movie database of information related to films including cast, writer, director, production company rating operated by admins and users.

The site enables registered users to submit new information and film related materials and edit the existing information. users are also invited to rate any film on a scale of 1 to 10 and the totals are converted into mean-rating that is displayed beside each title.

anyone can view the film information. a registration process is necessary however to contribute information to the site.

2 Project Features

The IMDB-UIU offers a rating scale that allows users to rate films on a scale of 1 to 10

The site indicates that submitted rating are filtered and weighted in various ways in order to produce a weighted mean that is displayed for each film.

Users can write comment for every movie in the comment section of the film.

The site top 50 list is a listing for the top rated 50 films based on rating by registered users.

Users can search movies by the title of the movie and can see the information of the movie.

3 Project ER Diagram

4 Project Schema Diagram

5 Source of Database

create table movie
(movie$_i dvarchar$(15),
title varchar(50),
release$_y earint$(4),
rating decimal(2,1),
vote int(10),
run$_t imeint$(10),
story$_l inetext$,
primary key (movie$_i d$)
);
create table person
(person$_i dvarchar$(15),
name varchar(50),
date$_o f_b irthdate$,
gender varchar(20),
primary key (person$_i d$)
);
create table genre
(genre$_i dvarchar$(15),
title varchar(50),
primary key (genre$_i d$)
);
create table language
(language$_i dvarchar$(15),
name varchar(50),
primary key (language$_i d$)
);
create table country
(country$_i dvarchar$(15),
name varchar(50),
primary key (country$_i d$)
);
create table location
(location$_i dvarchar$(15),
name varchar(50),
primary key (location$_i d$)
);
create table production
(production$_i dvarchar$(15),
title varchar(50),
primary key (production$_i d$)
);
create table movie$_d irector$
(movie$_i dvarchar$(15),
person$_i dvarchar$(15),
primary key (movie$_i d$),
foreign key (movie$_i d) references movie(movie$_i d) on delete cascade,
foreign key (person$_i d) references person(person$_i d) on delete cascade
);
create table movie$_w riter$
(movie$_i dvarchar$(15),
person$_i dvarchar$(15),
primary key (movie$_i d$),
foreign key (movie$_i d) references movie(movie$_i d) on delete cascade,
foreign key (person$_i d) references person(person$_i d) on delete cascade
);

create table movie$_c ast$
(movie$_i dvarchar(15)$,
person$_i dvarchar(15)$,
primary key (movie$_i d, person_i d$),
foreign key (movie$_i d) references movie(movie_i d) on delete cascade$,
foreign key (person$_i d) references person(person_i d) on delete cascade$
);
create table movie$_g enre$
(movie$_i dvarchar(15)$,
genre$_i dvarchar(15)$,
primary key (movie$_i d, genre_i d$),
foreign key (movie$_i d) references movie(movie_i d) on delete cascade$,
foreign key (genre$_i d) references genre(genre_i d) on delete cascade$
);
create table movie$_l anguage$
(movie$_i dvarchar(15)$,
language$_i dvarchar(15)$,
primary key (movie$_i d$),
foreign key (movie$_i d) references movie(movie_i d) on delete cascade$,
foreign key (language$_i d) references language(language_i d) on delete cascade$
);
create table movie$_c ountry$
(movie$_i dvarchar(15)$,
country$_i dvarchar(15)$,
primary key (movie$_i d$),
foreign key (movie$_i d) references movie(movie_i d) on delete cascade$,
foreign key (country$_i d) references country(country_i d) on delete cascade$
);
create table movie$_l ocation$
(movie$_i dvarchar(15)$,
location$_i dvarchar(15)$,
primary key (movie$_i d, location_i d$),
foreign key (movie$_i d) references movie(movie_i d) on delete cascade$,
foreign key (location$_i d) references location(location_i d) on delete cascade$
);
create table country$_l ocation$
(location$_i dvarchar(15)$,
country$_i dvarchar(15)$,
primary key (location$_i d$),
foreign key (location$_i d) references location(location_i d) on delete cascade$,
foreign key (country$_i d) references country(country_i d) on delete cascade$
);
create table movie$_p roduce$
(movie$_i dvarchar(15)$,
production$_i dvarchar(15)$,
foreign key (movie$_i d) references movie(movie_i d) on delete cascade$,
foreign key (production$_i d) references production(production_i d) on delete cascade$
);

6 Screenshots

7 Conclusions

The IMDB-UIU is a very user friendly site that can help a user a lot. the site is very important to know the information of a film. it can help a movie lover users a lot. the site is very important to daily life of a movie fan. it can help a film industry also. a production company or a film industry can also use the data from this site.

last of all we can tell that the web site is very efficient and use for any user. we want to develop the site gradually. so stay connected and support us to make a such kind of site.

thank you all.....