**Курсовой проект.**

**Текстовое описание БД и решаемых ею задач;**

База данных для хранения информации сайта [www.kinopoisk.ru](http://www.kinopoisk.ru/).

Задачи, хранение данных:

1. Данные пользователей, страна, город, пол, дата рождения, ФИО

2. Сериалы, наименование, количество эпизодов

3. Коментарии пользователей

4.

Примеры таблиц: https://www.kinopoisk.ru/media/video/4003777/

**Выводим список баз данных**

SHOW databases;

**Создаем базу данных если она отсутствует**

CREATE database IF not exists tester3 character set utf8 collate utf8\_general\_ci;

**Установка текущей базы данных**

USE tester3;

**Скрипты создания таблиц в базе данных**

SHOW databases;

CREATE database IF not exists tester3 character set utf8 collate utf8\_general\_ci;

USE tester3;

**# Таблица страны**

DROP TABLE IF EXISTS country;

CREATE TABLE IF NOT EXISTS country (  
 country\_id INT PRIMARY KEY,  
 name\_country VARCHAR(255)  
) ENGINE=INNODB;

INSERT INTO country VALUES (1,"Россия");  
INSERT INTO country VALUES (2,"США");  
INSERT INTO country VALUES (3,"Китай");  
  
SELECT \* FROM country;  
  
**# Таблица Пол пользователя**  
DROP TABLE IF EXISTS gender;  
CREATE TABLE IF NOT EXISTS gender (  
 gender\_id INT1 PRIMARY KEY,  
 name\_gender VARCHAR(255)  
) ENGINE=INNODB;  
  
INSERT INTO gender VALUES (1,"Мужской");  
INSERT INTO gender VALUES (2,"Женский");  
INSERT INTO gender VALUES (3,"Неопределился");  
  
SELECT \* FROM gender;  
  
**# Таблица Пользователи**  
DROP TABLE IF EXISTS users;  
CREATE TABLE IF NOT EXISTS users (  
 user\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 country\_id INT,  
 city VARCHAR(255),  
 gender int1,  
 name VARCHAR(255),  
 birthday DATE,  
 description TEXT,  
 priority TINYINT NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (country\_id)   
 REFERENCES country (country\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE,  
 FOREIGN KEY (gender)   
 REFERENCES gender (gender\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE  
) ENGINE=INNODB;  
  
INSERT INTO users VALUES (1,1,"Красноярск",1,"Иван Иванов",null,"Пользователь",1,null);  
INSERT INTO users VALUES (2,1,"Москва",1,"Петр Петров",null,"Пользователь 2",1,null);  
INSERT INTO users VALUES (3,1,"Москва",1,"Сидр Сидоров",null,"Пользователь 3",1,null);  
  
SELECT \* FROM users;  
  
SELECT user\_id,country.name\_country,gender.name\_gender,name,birthday,description,priority,created\_at   
 FROM users  
 LEFT JOIN country ON users.country\_id = country.country\_id  
 LEFT JOIN gender ON users.gender = gender.gender\_id;

**# Вызов хранимой процедуры**  
CALL get\_users;

**#https://www.kinopoisk.ru/media/video/4003777/comment/2429062/#comments**  
#  
CREATE TABLE IF NOT EXISTS media\_video (  
 media\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 title VARCHAR(255),  
 description TEXT,  
 embed VARCHAR(255),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
) ENGINE=INNODB;  
  
**# https://www.kinopoisk.ru/media/article/4003784/**  
CREATE TABLE IF NOT EXISTS media\_article (  
 article\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 title VARCHAR(255),  
 description TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
) ENGINE=INNODB;  
  
**# https://www.kinopoisk.ru/media/news/4003813/**  
CREATE TABLE IF NOT EXISTS media\_news (  
 news\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 title VARCHAR(255),  
 description TEXT,  
 istochnik VARCHAR(255),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
) ENGINE=INNODB;  
  
**# Таблица с ключевыми словами**  
CREATE TABLE IF NOT EXISTS keywords (  
 keyword\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 word VARCHAR(255)  
) ENGINE=INNODB;  
  
**# https://www.kinopoisk.ru/series/1367894/  
# https://www.kinopoisk.ru/series/1367892/  
# https://www.kinopoisk.ru/series/77044/**  
DROP TABLE IF EXISTS series\_keywords,recensii,episodes,series;  
  
**# Таблица с сериалами**  
CREATE TABLE IF NOT EXISTS series (  
 series\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 title VARCHAR(255),  
 description TEXT,  
 god INT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
) ENGINE=INNODB;  
# Добавление данных в таблицу  
  
**# Список ключевых слов для сериала**  
CREATE TABLE IF NOT EXISTS series\_keywords (  
 series\_id INT,  
 keyword\_id INT,  
 PRIMARY KEY (series\_id , keyword\_id),  
 FOREIGN KEY (series\_id)   
 REFERENCES series (series\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE,  
 FOREIGN KEY (keyword\_id)   
 REFERENCES keywords (keyword\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE  
) ENGINE=INNODB;  
  
**# Таблица с эпизодами сериала**  
DROP TABLE IF EXISTS episodes;  
**# https://www.kinopoisk.ru/film/77044/episodes/**  
CREATE TABLE IF NOT EXISTS episodes (  
 num\_episode INT,  
 num\_season INT,  
 series\_id INT,  
 title\_rus VARCHAR(255),  
 title\_eng VARCHAR(255),  
 episode\_date date,  
 god INT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (num\_episode,num\_season,series\_id),  
 FOREIGN KEY (series\_id)   
 REFERENCES series (series\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE  
) ENGINE=INNODB;  
  
  
**# Рецензии на сериалы**  
DROP TABLE IF EXISTS recensii;  
CREATE TABLE IF NOT EXISTS recensii (  
 recensii\_id INT AUTO\_INCREMENT,  
 series\_id INT,  
 description TEXT,  
 user\_id INT NOT NULL,  
 num\_like TINYINT NOT NULL,  
 num\_dislike TINYINT NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (recensii\_id , user\_id),  
 FOREIGN KEY (user\_id)   
 REFERENCES users (user\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE,  
 FOREIGN KEY (series\_id)   
 REFERENCES series (series\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE  
) ENGINE=INNODB;  
  
DROP TABLE IF EXISTS comments;  
CREATE TABLE IF NOT EXISTS comments (  
 comment\_id INT AUTO\_INCREMENT,  
 media\_id INT,  
 article\_id INT,  
 parent\_id INT,  
 news\_id INT,  
 description TEXT,  
 user\_id INT NOT NULL,  
 priority TINYINT NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (comment\_id , user\_id),  
 FOREIGN KEY (user\_id)   
 REFERENCES users (user\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE,  
 FOREIGN KEY (media\_id)   
 REFERENCES media\_video (media\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE,  
 FOREIGN KEY (article\_id)   
 REFERENCES media\_article (article\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE,  
 FOREIGN KEY (news\_id)   
 REFERENCES media\_news (news\_id)   
 ON UPDATE RESTRICT   
 ON DELETE CASCADE  
) ENGINE=INNODB;  
  
INSERT INTO series VALUES (77044,"Друзья (сериал 1994 – 2004)","Великий комедийный сериал, с героями которого мы и правда успели подружиться",1994,null);  
INSERT INTO series VALUES (77046,"Секретные материалы (сериал 1993 – 2018)","The X Files",1993,null);  
INSERT INTO series VALUES (77051,"Идиот (сериал 2003)","",2003,null);  
INSERT INTO episodes VALUES (1,1,77051,"Эпизод 1","Эпизод #1.1",null,2013,null);  
INSERT INTO episodes VALUES (2,1,77051,"Эпизод 2","Эпизод #1.2",null,2013,null);  
INSERT INTO episodes VALUES (3,1,77051,"Эпизод 3","Эпизод #1.3",null,2013,null);  
INSERT INTO episodes VALUES (4,1,77051,"Эпизод 4","Эпизод #1.4",null,2013,null);  
INSERT INTO episodes VALUES (5,1,77051,"Эпизод 5","Эпизод #1.5",null,2013,null);  
INSERT INTO episodes VALUES (6,1,77051,"Эпизод 6","Эпизод #1.6",null,2013,null);  
INSERT INTO episodes VALUES (7,1,77051,"Эпизод 7","Эпизод #1.7",null,2013,null);  
INSERT INTO episodes VALUES (8,1,77051,"Эпизод 8","Эпизод #1.8",null,2013,null);  
INSERT INTO episodes VALUES (9,1,77051,"Эпизод 9","Эпизод #1.9",null,2013,null);  
INSERT INTO episodes VALUES (10,1,77051,"Эпизод 10","Эпизод #1.10",null,2013,null);  
SELECT \* FROM series;  
  
  
## Вывести количество пользоваетелей  
CALL count\_users();  
## ВЫЗОВ ХРАНИМОЙ ПРОЦЕДУРЫ С ПАРАМЕТРОМ  
CALL proc\_in(100);  
CALL proc\_in(200);  
CALL proc\_in(300);  
## ВЫЗОВ ХРАНИМОЙ ПРОЦЕДУРЫ С ВОЗВРАТОМ ПАРАМЕТРА  
CALL proc\_out(@var2);  
SELECT @var2;  
# Вывести количество эпизодов для сериала с ID 77051  
CALL get\_episodes\_count(77051);  
# Вывести список сериалов 0 - смещение от начала, 2 - количество в выборке  
CALL get\_series(0,2)

