-- Создать при помощи SQL несколько таблиц

-- Например, таблица “Фильмы” и таблица “Киностудии”

-- Заполнить таблицы с данными в соответствии с выбранными типами

-- Продемонстрировать работу с

-- операторами переименования таблиц\полей,

-- операторами добавления и удаления таблиц\полей;

-- добавления ключей(внешних);

-- оператором псевдонима;

-- оператором CASE;

-- оператором IF;

-- Подумать и описать проблемы своего проектирования базы данных;

-- Оформить работу так, чтобы результатами было пользоваться удобно не только лично вам.

use lesson02;

DROP TABLE movies;

CREATE TABLE movies

(

movies\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

movies\_title VARCHAR(45),

movies\_year INT,

movies\_genre VARCHAR(45)

);

INSERT movies(movies\_title, movies\_year, movies\_genre)

VALUES

('Какой-то фильм',2022,'Комедия'),

('Гарри Поттер', 2010, 'Фантастика'),

('Мстители', 2015, 'Фантастика'),

('Человек паук', 2005, "фантастика");

SELECT \* FROM movies;

SELECT

movies\_id as '№',

movies\_title as 'Наименование',

movies\_year as 'Год издания',

movies\_genre as 'Жанр'

FROM movies;

-- Операции с таблицами

-- 1. Переименование делается с помощью команды RENAME TABLE.

-- RENAME TABLE old\_name TO new\_name;

RENAME TABLE movies TO films;

SELECT \* FROM films;

-- 2. Для добавления нового столбца нам понадобится команда ADD.

-- ALTER TABLE table\_name

-- ADD column\_name VARCHAR(50) NULL;

ALTER TABLE films

ADD films\_rating INT NULL;

SELECT \* FROM films;

-- 2.1 Обновить запись

-- UPDATE table\_name

-- SET column1 = value1, column2 = value2, ...

-- WHERE condition;

use lesson01;

UPDATE films

SET films\_rating = 5

WHERE TRUE;

UPDATE films

SET films\_rating = 1

WHERE movies\_id = 1;

UPDATE films

SET films\_rating = 4

WHERE movies\_id = 2;

UPDATE films

SET films\_rating = 2

WHERE movies\_id = 3;

SELECT \* FROM films;

-- 3. Чтобы удалить столбец в MySQL используется ALTER TABLE с DROP COLUMN:

-- ALTER TABLE table\_name

-- DROP COLUMN column\_name;

ALTER TABLE films

DROP COLUMN films\_rating;

-- Внешние ключи

DROP TABLE distribution;

CREATE TABLE distribution

(

studio\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

studio\_title VARCHAR(100)

);

INSERT distribution(studio\_title)

VALUES

('Неизвестная студия'),

('Paramount Pictures'),

('Мосфильм'),

('The Walt Disney Company');

SELECT \* FROM distribution;

ALTER TABLE films

ADD creator INT NOT NULL DEFAULT 1;

SELECT \* FROM films;

-- ADD FOREIGN KEY ()

SELECT

\*

FROM films LEFT JOIN distribution

ON films.creator = distribution.studio\_id;

ALTER TABLE films

ADD FOREIGN KEY (creator) REFERENCES distribution(studio\_id);

SELECT \* FROM films;

-- Удаление данных

SELECT \* FROM films;

TRUNCATE TABLE films;

INSERT films(movies\_title, movies\_year, movies\_genre,creator )

VALUES

('Гарри Поттер', 2010, 'Фантастика',1),

('Мстители', 2015, 'Фантастика',1),

('Человек паук', 2005, 'фантастика',1);

-- CASE

SELECT \* FROM films;

SELECT

movies\_title,

films\_rating

FROM films;

SELECT

movies\_title as 'Наименование',

films\_rating as 'Ретинг',

CASE films\_rating

WHEN 1 THEN '⭐️'

WHEN 2 THEN '⭐️⭐️'

WHEN 3 THEN '⭐️⭐️⭐️'

WHEN 4 THEN '⭐️⭐️⭐️⭐️'

WHEN 5 THEN '⭐️⭐️⭐️⭐️⭐️'

ELSE '⭐︎⭐︎⭐︎⭐︎⭐︎'

END AS 'Рейтинг'

FROM films;

-- Ещё

SELECT

movies\_title as 'Наименование',

films\_rating as 'Рейтинг',

CASE

WHEN films\_rating >= 3 THEN 'Рекомендуем к просмотру'

/\* WHEN films\_rating = NULL THEN 'Нет просмотров' --

-- Проверить значение на равенство NULL

-- c помощью обычных арифметических

-- операторов сравнения (=, < или <>) нельзя.

-- Вместо этого нужно пользоваться операторами IS NULL и IS NOT NULL

\*/

WHEN films\_rating IS NULL THEN 'Нет просмотров'

ELSE 'Скука'

END AS 'Оценочное мнение'

FROM films;

SELECT

movies\_title as 'Наименование',

films\_rating as 'Ретинг',

CASE

WHEN films\_rating >= 1 AND films\_rating <=2 THEN 'Не рекомендуем к просмотру'

WHEN films\_rating IS NULL THEN 'Нет просмотров'

ELSE 'Нормальная тема'

END AS 'Оценочное мнение'

FROM films;

SELECT

movies\_title as 'Наименование',

IF (films\_rating >= 3, 'Смотрим','Не смотрим') AS 'Рекомендация'

FROM films;

-- Дополнительно

SELECT

UTC\_TIME(),

CURRENT\_TIME(),

HOUR(CURRENT\_TIME()),

IF (HOUR(CURRENT\_TIME() - 1) <= 12, 'Утро добрым не бывает','Работаем');