ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

КУРСОВАЯ РАБОТА

По дисциплине «Информационное обеспечение систем управления»

Выполнил: ст. гр. ТУУ – 411

Виноградов М.А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Цель курсового проекта 2](#_Toc91076797)

[Задания на курсовой проект 2](#_Toc91076798)

[Инфологическое проектирование БД 2](#_Toc91076799)

[Логическое проектирование реляционной БД 4](#_Toc91076800)

[Составление реляционных отношений 5](#_Toc91076801)

[Нормализация полученных отношений 9](#_Toc91076802)

[Физическое проектирование БД 9](#_Toc91076803)

[Разработка необходимых запросов 13](#_Toc91076804)

[Вывод 16](#_Toc91076805)

## **Цель курсового проекта**

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server.

## Задания на курсовой проект

В данном курсовом проекте ставится задача разработать РБД в СУБД Microsoft SQL Server по заданной теме (формализующую заданную предметную область (ПрО). Проектирование РБД проводится с помощью метода «Сущность-связь». Проверка построенной модели РБД осуществляется с помощью метода нормализации отношений. Пояснительная записка должна содержать пункты по проектированию РБД и пункты по разработке РБД в СУБД Microsoft SQL Server: разработка скриптов на создание и заполнение РБД, разработка необходимых функций, процедур, триггеров и представлений (views).

Пояснительная записка оформляется согласно ГОСТ 7.32-2017 «ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ» [1].

## Инфологическое проектирование БД

Описание предметной области. БД создается для информационного обслуживания сотрудников и посетителей магазина. Магазин продает различные виды фильмов (на носителях различного вида).

Готовые запросы:

1. Найти фильм по названию (по жанру, исполнителям, режиссеру).
2. Показывать список возможных носителей для выбранного фильма.
3. Показывать список фильмов по жанрам.
4. Показывать информацию по выбранному фильму (жанр, исполнители, режиссер, краткое содержание).
5. Показывать список наиболее продаваемых фильмов.
6. Показывать количество проданных фильмов за отчетный период (сумму продаж).

Выделим базовые сущности этой ПрО:

Фильм – Название, описание, продолжительность, режисёр, страна производства, жанр.

Режиссёр – Имя, фамилия, дата рождения, дата смерти.

Старана производства – Название.

Жанр – Название.

Актёрский состав – Фильм, актёр.

Актёр – Имя, фамилия, дата рождения, дата смерти.

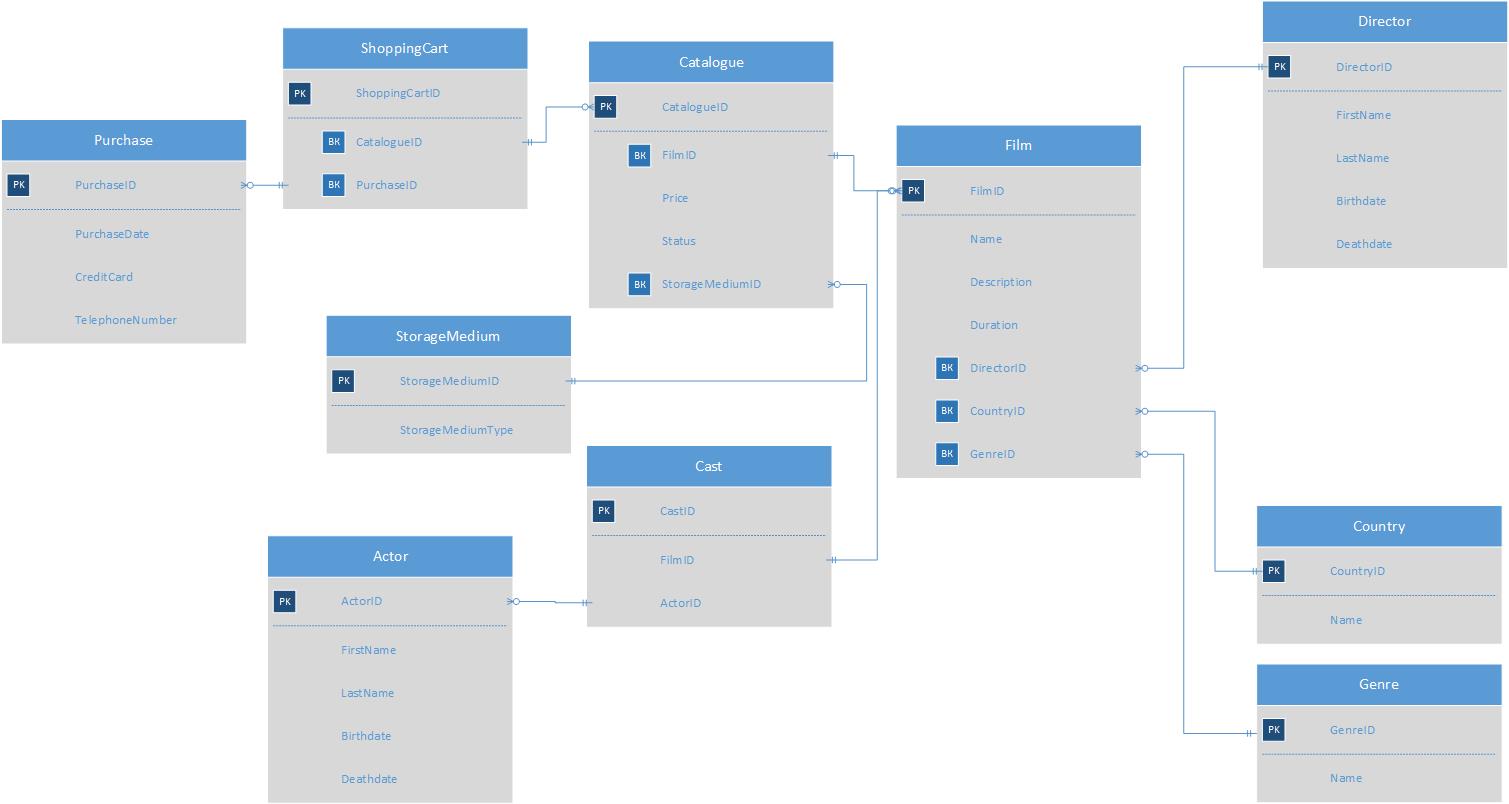
Каталог – Фильм, цена, статус, носитель.

Носитель – Название носителя.

Корзина покупок – Предмет из каталога, покупка.

Покупка – Дата покупки, кредитная карта, телефонный номер.

## Логическое проектирование реляционной БД



**Рисунок 1** – ER диаграмма будущей БД

## Составление реляционных отношений

Таблица 1 – Схема отношения Фильм (Film)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | FilmID | Целое число | первичный ключ |
| Название | Name | Строка (255) | обязательное поле |
| Описание | Description | Строка (255) | обязательное поле |
| Продожительность | Duration | Целое число | обязательное поле |
| Страна производства | CountryID | Целое число | внешний ключ (Country), обязательное поле |
| Жанр | GenreID | Целое число | внешний ключ (Genre), обязательное поле |
| Режиссёр | DirectorID | Целое число | внешний ключ (Director), обязательное поле |

Таблица 2 – Схема отношения режиссёр (Director)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | DirectorID | Целое число | первичный ключ |
| Имя | FirstName | Строка (255) | обязательное поле |
| Фамилия | LastName | Целое число | Обязательное поле |
| Дата рождения | Birthdate | Целое число | Обязательное поле |
| Дата смерти | Deathdate | Целое число |  |

Таблица 3 – Схема отношения носитель (StorageMedium)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | StorageMediumID | Целое число | первичный ключ |
| Название носителя | StorageMediumType | Строка (255) | обязательное поле |

Таблица 4 – Схема отношения страна (Country)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | CountryID | Целое число | первичный ключ |
| Название страны | Name | Строка (255) | обязательное поле |

Таблица 5 – Схема отношения актёр (Actor)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ActorID | Целое число | первичный ключ |
| Имя | FirstName | Строка (255) | обязательное поле |
| Фамилия | LastName | Строка (255) | обязательное поле |
| Дата рождения | Birthdate | Дата | обязательное поле |
| Дата смерти | Deathdate | Дата |  |

Таблица 6 – Схема отношения Актёрский состав (Cast)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | CastID | Целое число | первичный ключ |
| Фильм | FilmID | Целое число | внешний ключ (Film), обязательное поле |
| Актёр | ActorID | Целое число | внешний ключ (Actor), обязательное поле |

Таблица 7 – Схема Каталог (Catalogue)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | CatalogueID | Целое число | первичный ключ |
| Фильм | FimlID | Целое число | внешний ключ (Film), обязательное поле |
| Цена | Price | Целое число | обязательное поле |
| Статус | Status | Бит | обязательное поле |
| Носитель | StorageMediumID | Целое число | внешний ключ (StorageMedium), обязательное поле |

Таблица 8 – Схема отношения Покупка (Purchase)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | PurchaseID | Целое число | первичный ключ |
| Название | PurchaseDate | Дата | обязательное поле |
| Кредитная карта | CreditCard | Строка (255) | обязательное поле |
| Телефонный номер | TelephoneNumber | Строка (255) | обязательное поле |

Таблица 9 – Схема причина корзина покупок (ShoppingCart)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ShoppingCartID | Целое число | первичный ключ |
| Предмет из каталога | CatalogueID | Целое число | обязательное поле |
| Покупка | PurchaseID | Целое число | Обязательное поле |

Таблица 10 – Схема жанр (Genre)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | GenreID | Целое число | первичный ключ |
| Название | Name | Строка (255) | обязательное поле |

## Нормализация полученных отношений

**1НФ.** Для приведения таблиц к 1НФ требуется составить таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.

Все отношения находятся в 1 НФ.

**2НФ.** Все отношения находятся во 2НФ, так как для каждого из них введен уникальный ключ.

## Физическое проектирование БД

Разработка скриптов на создание базы данных и таблиц:

InitDB – создаёт БД FilmShop

CREATE DATABASE FilmShop;

InitTable – создаёт таблицы.

USE FilmShop;

CREATE TABLE Director (

DirectorID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

Birthdate DATE NOT NULL,

Deathdate DATE DEFAULT NULL,

);

CREATE TABLE StorageMedium (

StorageMediumID INT PRIMARY KEY IDENTITY(1,1),

StorageMediumType VARCHAR(255) NOT NULL,

);

CREATE TABLE Country (

CountryID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE Genre (

GenreID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

CREATE TABLE Film (

FilmID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

[Description] VARCHAR(255) NOT NULL,

Duration INT NOT NULL,

CountryID INT NOT NULL,

FOREIGN KEY (CountryID) REFERENCES Country (CountryID),

GenreID INT NOT NULL,

FOREIGN KEY (GenreID) REFERENCES Genre (GenreID),

DirectorID INT NOT NULL,

FOREIGN KEY (DirectorID) REFERENCES Director (DirectorID),

);

CREATE TABLE Actor (

ActorID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

Birthdate DATE NOT NULL,

Deatdate DATE DEFAULT NULL,

);

CREATE TABLE [Cast] (

CastID INT PRIMARY KEY IDENTITY(1,1),

FilmID INT,

FOREIGN KEY (FilmID) REFERENCES Film (FilmID),

ActorID INT,

FOREIGN KEY (ActorID) REFERENCES Actor (ActorID),

);

CREATE TABLE Catalogue (

CatalogueID INT PRIMARY KEY IDENTITY(1,1),

FilmID INT,

FOREIGN KEY (FilmID) REFERENCES Film (FilmID),

Price INT NOT NULL,

[Status] BIT NOT NULL,

StorageMediumID INT,

FOREIGN KEY (StorageMediumID) REFERENCES StorageMedium (StorageMediumID),

);

CREATE TABLE Purchase (

PurchaseID INT PRIMARY KEY IDENTITY(1,1),

PurchaseDate DATE NOT NULL,

CreditCard VARCHAR(255) NOT NULL,

TelephoneNumber VARCHAR(255) NOT NULL,

);

CREATE TABLE ShoppingCart (

ShoppingCartID INT PRIMARY KEY IDENTITY(1,1),

CatalogueID INT NOT NULL,

FOREIGN KEY (CatalogueID) REFERENCES Catalogue (CatalogueID),

PurchaseID INT NOT NULL,

FOREIGN KEY (PurchaseID) REFERENCES Purchase (PurchaseID),

);

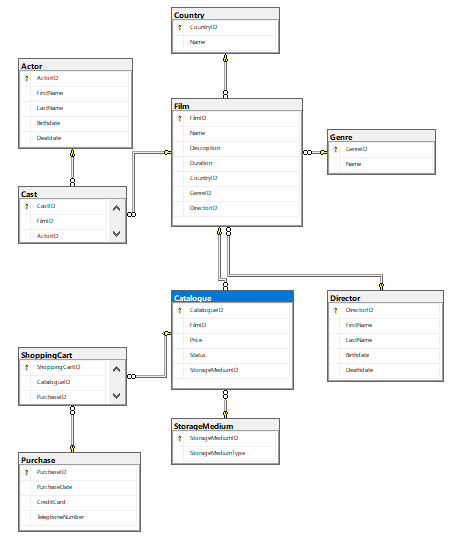


Рисунок 2 – ER диаграмма сгенерированная MS Management Studio

Разработка скриптов на добавление данных в таблицы

-- Insert Directors

INSERT INTO Director VALUES ('Quentin', 'Tarantino', '1963-03-27', NULL);

INSERT INTO Director VALUES ('Stanley', 'Kubrick', '1928-07-26', '1999-03-07');

-- Insert Storage Medium

INSERT INTO StorageMedium VALUES ('CD');

INSERT INTO StorageMedium VALUES ('Tape');

INSERT INTO StorageMedium VALUES ('Online');

-- Insert Countries

INSERT INTO Country VALUES ('United States');

INSERT INTO Country VALUES ('United Kingdom');

-- Insert Genres

INSERT INTO Genre VALUES ('Action');

INSERT INTO Genre VALUES ('Adventure');

INSERT INTO Genre VALUES ('Drama');

INSERT INTO Genre VALUES ('Thriller');

--Insert Film

INSERT INTO Film VALUES ('Pulp Fiction', 'Cool','120',1,1,1);

INSERT INTO Film VALUES ('Django Unchained', 'Piw pow','150',1,1,1);

INSERT INTO Film VALUES ('A Clockwork Orange', 'Wow','152',2,2,2);

--Insert Actors

INSERT INTO Actor VALUES ('John', 'Travolta','1954-02-18', NULL);

INSERT INTO Actor VALUES ('Samuel', 'L.Jackson','1948-12-21', NULL);

INSERT INTO Actor VALUES ('Malcolm', 'McDowell','1943-06-13', NULL);

--Insert Cast

INSERT INTO [Cast] VALUES (1,1);

INSERT INTO [Cast] VALUES (1,2);

INSERT INTO [Cast] VALUES (2,2);

INSERT INTO [Cast] VALUES (3,3);

--Insert Catologue

INSERT INTO Catalogue VALUES (1,10,1,1);

INSERT INTO Catalogue VALUES (1,10,1,2);

INSERT INTO Catalogue VALUES (2,11,1,1);

INSERT INTO Catalogue VALUES (3,15,1,2);

INSERT INTO Catalogue VALUES (1,10,0,1);

INSERT INTO Catalogue VALUES (1,10,0,2);

INSERT INTO Catalogue VALUES (1,5,0,3);

--Insert Purchase

INSERT INTO Purchase VALUES ('1999-04-12','1234 3242 4234 3243', '5435323443');

INSERT INTO Purchase VALUES ('2000-04-12','1234 3242 4234 4234', '5435322342');

--Insert ShoppingCart

INSERT INTO ShoppingCart VALUES (1,1);

INSERT INTO ShoppingCart VALUES (2,1);

INSERT INTO ShoppingCart VALUES (3,2);

INSERT INTO ShoppingCart VALUES (4,2);

## Разработка необходимых запросов

Найти фильм по названию

--1. Найти фильм по названию  
SELECT \* FROM Film WHERE [Name] = 'Pulp fiction'



Рисунок 3 ‑ Результат выполнения запроса

Найти фильм по жанру.

--1. Найти фильм по жанру.  
SELECT a.\* FROM Film as a   
JOIN Genre as b ON a.GenreID = b.GenreID  
WHERE b.Name = 'Action'

  
Рисунок 4 ‑ Результат выполнения запроса

Найти фильм по исполнителю.

--1. Найти фильм по исполнителям.  
SELECT a.\* FROM Film as a  
JOIN [Cast] as b ON a.FilmID = b.FilmID  
JOIN Actor as c ON b.ActorID = c.ActorID  
WHERE c.FirstName = 'John'

  
Рисунок 5 ‑ Результат выполнения запроса

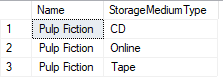
Найти фильм по режиссёру.

--1. Найти фильм по режисёру.  
SELECT a.\* FROM Film as a  
JOIN Director as b ON a.DirectorID = b.DirectorID  
WHERE b.FirstName = 'Quentin'

  
Рисунок 6 ‑ Результат выполнения запроса

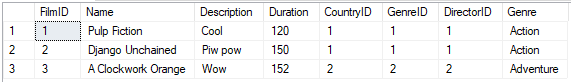
Показывать список возможных носителей для выбранного фильма.

--2. Показывать список возможных носителей для выбранного фильма.  
SELECT DISTINCT c.Name, b.StorageMediumType FROM Catalogue as a  
JOIN StorageMedium as b ON a.StorageMediumID = b.StorageMediumID  
JOIN Film as c ON a.FilmID = c.FilmID  
WHERE c.[Name] = 'Pulp Fiction'

  
Рисунок 7 ‑ Результат выполнения запроса

Показывать список фильмов по жанрам.

--3. Показывать список фильмов по жанрам.  
SELECT a.\*, b.Name as Genre FROM Film as a  
JOIN Genre as b ON a.GenreID = b.GenreID

  
Рисунок 8 ‑ Результат выполнения запроса

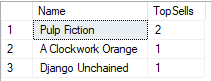
Показывать информацию по выбранному фильму.

--4. Показывать информацию по выбранному фильму (жанр, исполнители, режиссер, краткое содержание).  
SELECT \* FROM Film  
WHERE Film.[Name] = 'Django unchained'

  
Рисунок 9 ‑ Результат выполнения запроса

Показывать список наиболее продаваемых фильмов.

--5. Показывать список наиболее продаваемых фильмов.  
SELECT c.[Name], COUNT(\*) as TopSells FROM ShoppingCart as a  
JOIN Catalogue as b ON a.CatalogueID = b.CatalogueID  
JOIN Film as c ON b.FilmID = c.FilmID  
GROUP BY c.Name|  
ORDER BY TopSells DESC

  
Рисунок 10 ‑ Результат выполнения запроса

Показывать количество проданных фильмов за отчетный период (сумму продаж).

--6. Показывать количество проданных фильмов за отчетный период (сумму продаж).  
SELECT SUM(c.Price) as Profit FROM Purchase as a  
JOIN ShoppingCart as b ON a.PurchaseID = b.PurchaseID  
JOIN Catalogue as c ON b.CatalogueID = c.CatalogueID  
WHERE YEAR(a.PurchaseDate) = 1999

  
Рисунок 11 ‑ Результат выполнения запроса

## Вывод

Изучил методы и закрепил знания в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server.