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

Гаврилов Е.А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Цель курсового проекта 2](#_Toc90326791)

[Задания на курсовой проект 2](#_Toc90326792)

[Инфологическое проектирование БД 2](#_Toc90326793)

[Логическое проектирование реляционной БД 4](#_Toc90326794)

[Составление реляционных отношений 5](#_Toc90326795)

[Нормализация полученных отношений 8](#_Toc90326796)

[Физическое проектирование БД 8](#_Toc90326797)

[Разработка необходимых запросов 13](#_Toc90326798)

[Вывод 15](#_Toc90326799)

## **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.я DELPHI и системе управления базами данных Paradox.

## 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 клиентов магазина. Покупатели самостоятельно выбирают и находят понравившийся им товар.

Готовые запросы:

1. Показать все произведения данного композитора (исполнителя), имеющиеся в магазине.
2. Показать местоположение выбранного произведения.
3. Показать список носителей для выбранного произведения.
4. Показать список произведений по жанру исполнения.
5. Находить произведение по названию, году выпуска, альбому и т.д.
6. Показывать список произведений данного композитора (исполнителя) по выбранным годам творчества.

Выделим базовые сущности этой ПрО:

Композиция – серийный номер, композитор, название, описание, продолжительность, год выпуска, альбом, носитель, страна производства.

Композитор – имя, день рождения.

Альбом – название альбома

Носитель – название носителя

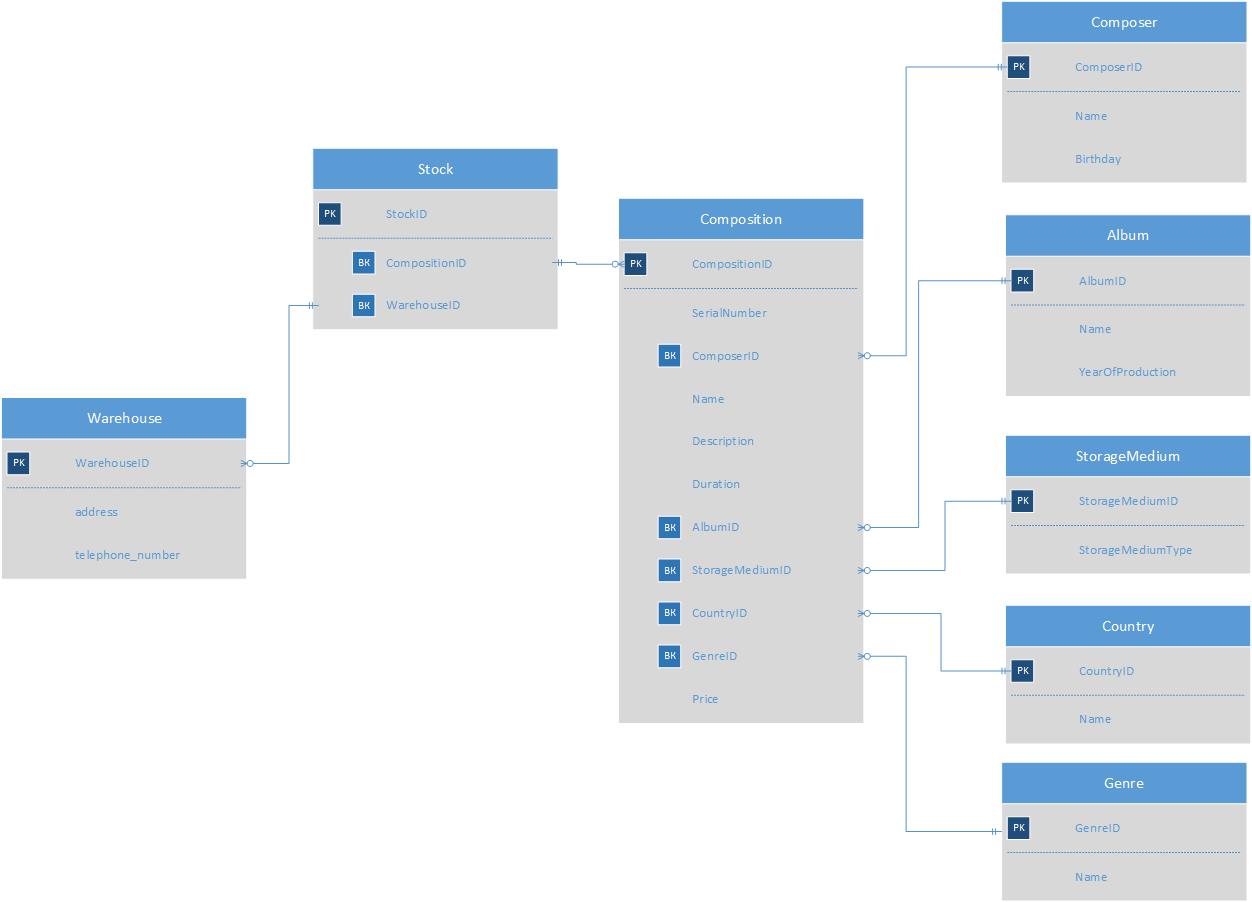
Страна производства – название страна производства

Жанр – название жанра музыки

Ассортимент – Композиция, склад где хранится данная композиция

Склад – Адрес склада, телефонный номер склада

## Логическое проектирование реляционной БД



**Рисунок 1** – ER диаграмма будущей БД

## Составление реляционных отношений

Таблица 1 – Схема отношения композиция (Composition)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | CompositionID | Целое число | первичный ключ |
| Серийный номер | SerialNumber | Строка (255) | обязательное поле |
| Композитор | ComposerID | Целое число | внешний ключ (Composer), обязательное поле |
| Название | Name | Строка (255) | обязательное поле |
| Продолжительность | Duration | Целое число | Обязательное поле |
| Альбом | AlbumID | Целой число | внешний ключ (Album), |
| Носитель | StorageMediumID | Целой число | внешний ключ (StorageMedium), обязательное поле |
| Страна производства | CountryID | Целой число | внешний ключ (Country), обязательное поле |
| Жанр | GenreID | Целой число | внешний ключ (Genre), обязательное поле |
| Цена | Price | Число с плавающей точкой | обязательное поле |

Таблица 2 – Схема отношения жанр (Genre)

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

Таблица 3 – Схема отношения страна производства (Country)

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

Таблица 4 – 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 5 – Схема отношения альбом (Album)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | AlbumID | Целое число | первичный ключ |
| Название альбома | Name | Строка (255) | обязательное поле |
| Описание | Description | Строка (255) |  |
| Год выпуска | DateOfProduction | Date |  |

Таблица 6 – Схема отношения исполнитель (Composer)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ComposerID | Целое число | первичный ключ |
| Имя, фамилия | Name | Строка (255) | обязательное поле |
| День рождения | Birthdate | Date |  |

Таблица 7 – Схема отношения склад (Warehouse)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | WarehouseID | Целое число | первичный ключ |
| Адрес | Name | Строка (255) | обязательное поле |
| Телефонный номер | TelephoneNumber | Строка (255) | обязательное поле |

Таблица 8 – Схема отношения ассортимент (Stock)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | StockID | Целое число | первичный ключ |
| Композиция | CompositionID | Целое число | внешний ключ (Composition), обязательное поле |
| Склад | WarehouseID | Целое число | внешний ключ (Warehouse), 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 MusicShop

CREATE DATABASE MusicShop;

InitTable – создаёт таблицы.

USE MusicShop;

Создание таблицы Composer

CREATE TABLE Composer (

ComposerID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

Birthday date,

);

Создание таблицы Album

CREATE TABLE Album (

AlbumID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

[Description] VARCHAR(255),

DateOfProduction date,

);

Создание таблицы StorageMedium

CREATE TABLE StorageMedium (

StorageMediumID INT PRIMARY KEY IDENTITY(1,1),

StorageMediumType VARCHAR(255) NOT NULL,

);

Создание таблицы Country

CREATE TABLE Country (

CountryID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

Создание таблицы Genre

CREATE TABLE Genre (

GenreID INT PRIMARY KEY IDENTITY(1,1),

[Name] VARCHAR(255) NOT NULL,

);

Создание таблицы Composition

CREATE TABLE Composition (

CompositionID INT PRIMARY KEY IDENTITY(1,1),

SerialNumber VARCHAR(255) NOT NULL,

ComposerID INT NOT NULL,

FOREIGN KEY (ComposerID) REFERENCES Composer (ComposerID),

[Name] VARCHAR(255) NOT NULL,

Duration INT NOT NULL,

AlbumID INT,

FOREIGN KEY (AlbumID) REFERENCES Album (AlbumID),

StorageMediumID INT NOT NULL,

FOREIGN KEY (StorageMediumID) REFERENCES StorageMedium (StorageMediumID),

CountryID INT NOT NULL,

FOREIGN KEY (CountryID) REFERENCES Country (CountryID),

GenreID INT NOT NULL,

FOREIGN KEY (GenreID) REFERENCES Genre (GenreID),

Price FLOAT NOT NULL,

);

Создание таблицы Warehouse

CREATE TABLE Warehouse (

WarehouseID INT PRIMARY KEY IDENTITY(1,1),

[Address] VARCHAR(255) NOT NULL,

TelephoneNumber VARCHAR(255) NOT NULL,

);

Создание таблицы Stock

CREATE TABLE Stock (

StockID INT PRIMARY KEY IDENTITY(1,1),

CompositionID INT NOT NULL,

FOREIGN KEY (CompositionID) REFERENCES Composition (CompositionID),

WarehouseID INT NOT NULL,

FOREIGN KEY (WarehouseID) REFERENCES Warehouse (WarehouseID),

);

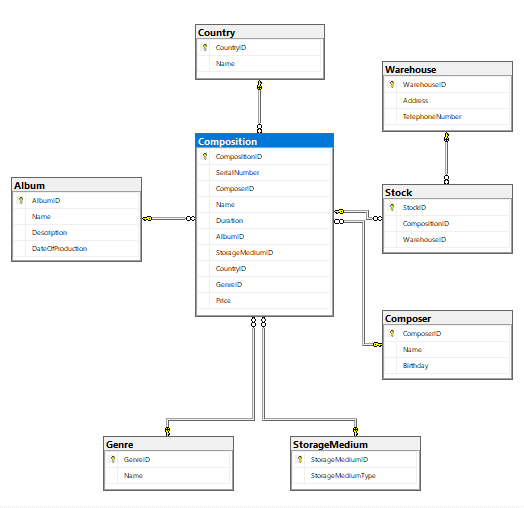


Рисунок 2 – ER диаграмма сгенерированная MS Management Studio

Разработка скриптов на добавление данных в таблицы

-- Insert Composers

INSERT INTO Composer VALUES ('Kanye West', '1977-07-08');

INSERT INTO Composer VALUES ('Frank Sinatra', '1915-12-12');

INSERT INTO Composer VALUES ('Jimi Hendrix', '1942-11-27');

-- Insert Albums

INSERT INTO Album VALUES ('Graduation','Cool', '2007-09-11');

INSERT INTO Album VALUES ('My Beautiful Dark Twisted Fantasy','Good', '2013-06-03');

INSERT INTO Album VALUES ('My Way','Rare', '1969-12-23');

INSERT INTO Album VALUES ('Are You Experienced', 'Usual', '1967-12-05');

-- 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 ('Spain');

-- Insert Genres

INSERT INTO Genre VALUES ('Pop');

INSERT INTO Genre VALUES ('Hard rock');

INSERT INTO Genre VALUES ('Rap');

-- Insert Compositions

INSERT INTO Composition VALUES ('4CE0460D0G',

1, --ComposerID

'Flashing Lights', --Name

241, --Duration

1, --AlbumID

1, --StorageMediumID

1, --CountryID

3, --GenreID

5 --Price

);

INSERT INTO Composition VALUES ('4CE0460D0G',

1, --ComposerID

'Flashing Lights', --Name

241, --Duration

1, --AlbumID

3, --StorageMediumID

1, --CountryID

3, --GenreID

5 --Price

);

INSERT INTO Composition VALUES ('4CE0460D0G',

1, --ComposerID

'Flashing Lights', --Name

241, --Duration

1, --AlbumID

2, --StorageMediumID

1, --CountryID

3, --GenreID

4 --Price

);

INSERT INTO Composition VALUES ('4CE0460D0G',

1, --ComposerID

'Power', --Name

201, --Duration

2, --AlbumID

1, --StorageMediumID

1, --CountryID

3, --GenreID

7 --Price

);

INSERT INTO Composition VALUES ('4CE0460DSG',

2, --ComposerID

'Yesterday', --Name

223, --Duration

3, --AlbumID

2, --StorageMediumID

1, --CountryID

1, --GenreID

10 --Price

);

INSERT INTO Composition VALUES ('4CE0443454',

3, --ComposerID

'Can You See Me', --Name

181, --Duration

4, --AlbumID

2, --StorageMediumID

1, --CountryID

2, --GenreID

12 --Price

);

--INSERT INTO Composition (SerialNumber ,ComposerID , [Name], Duration, StorageMediumID, CountryID, GenreID, Price)

--VALUES (1, 1, 'name', 12,1,1,1,13);

INSERT INTO Warehouse VALUES ('ul. Kosmonavtov 35-11,Vyborg,Leningradskaya oblast','89100900602');

INSERT INTO Warehouse VALUES ('Sheldon Cooper & Leonard Hofstadter,Apt 4B 2311 North Los Robles Avenue,Pasadena'

,'89100900703');

INSERT INTO Stock VALUES (1,1);

INSERT INTO Stock VALUES (1,1);

INSERT INTO Stock VALUES (1,1);

INSERT INTO Stock VALUES (2,2);

INSERT INTO Stock VALUES (3,1);

INSERT INTO Stock VALUES (4,1);

INSERT INTO Stock VALUES (1,2);

INSERT INTO Stock VALUES (5,1);

INSERT INTO Stock VALUES (2,1);

INSERT INTO Stock VALUES (2,1);

INSERT INTO Stock VALUES (3,2);

INSERT INTO Stock VALUES (2,2);

INSERT INTO Stock VALUES (2,1);

## Разработка необходимых запросов

Показать все произведения данного композитора (исполнителя), имеющиеся в магазине.

--1. Показать все произведения данного композитора (исполнителя), имеющиеся в магазине;  
SELECT DIS TINCT b.[Name] FROM Stock as a  
JOIN Composition as b ON a.CompositionID = b.CompositionID   
WHERE ComposerID = (SELECT ComposerID FROM Composer WHERE [Name] = 'Kanye West')



Рисунок 3 ‑ Результат выполнения запроса

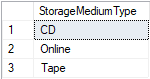
Показать местоположение выбранного произведения.

--2. Показать местоположение выбранного произведения;  
SELECT [Address] FROM Warehouse as a  
JOIN Stock as b ON a.WarehouseID = b.WarehouseID  
WHERE b.CompositionID = (SELECT CompositionID FROM Composition WHERE [Name] = 'Yesterday')

  
Рисунок 4 ‑ Результат выполнения запроса

Показать местоположение выбранного произведения.

--3. Показать список носителей для выбранного произведени  
SELECT DISTINCT StorageMedium.StorageMediumType FROM Composition, StorageMedium  
WHERE Composition.[Name] = 'Flashing Lights' AND Composition.StorageMediumID = StorageMedium.StorageMediumID

  
Рисунок 5 ‑ Результат выполнения запроса

Показать список произведений по жанру исполнения.

--4. Показать список произведений по жанру исполнения;  
SELECT DISTINCT Composition.[Name] FROM Composition, Genre  
WHERE Genre.[Name] = 'Rap' AND Genre.GenreID = Composition.GenreID

  
Рисунок 6 ‑ Результат выполнения запроса

Находить произведение по названию, году выпуска, альбому и т.д.

----5. Находить произведение по названию, году выпуска, альбому и т.д.  
SELECT Composition.[Name], Composition.[Name], Album.[Name] FROM Composition, Album  
WHERE Composition.[Name] = 'Power' AND Composition.AlbumID = Album.AlbumID  
AND Album.[Name] = 'My Beautiful Dark Twisted Fantasy' AND YEAR(Album.DateOfProduction) = 2013

  
Рисунок 7 ‑ Результат выполнения запроса

Находить произведение по названию, году выпуска, альбому и т.д.

--6. Показывать список произведений данного композитора (исполнителя) по выбранным годам творчества.  
SELECT DISTINCT Composition.[Name] FROM Composition  
JOIN Album ON Album.AlbumID = Composition.AlbumID  
JOIN Composer ON Composer.ComposerID = Composition.ComposerID  
WHERE YEAR(Album.DateOfProduction) BETWEEN 2000 AND 2015 AND Composer.[Name] = 'Kanye West'

  
Рисунок 8 ‑ 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.