

**ОТЧЕТ ПО ИНДИВИДУАЛЬНОМУ ЗАДАНИЮ:
РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗАЦИИ ДЛЯ ПУНКТА ПРОКАТА
ВИДЕОКАССЕТ**

ВЫПОЛНИЛ: СТУДЕНТКА ГРУППЫ ВВО-19 РЕДЬКИНА Я.Б.
СДАЧА: 25.12.2021 Г.

ОГЛАВЛЕНИЕ

1. Описание системы
 - Сущности
 - Связи между сущностями
 - Функциональность системы
 - Требования к поддержке целостности данных
 - Другие возможности системы
2. Структура системы
 - Таблицы и типы данных
 - Примеры данных для таблиц
 - Связи таблиц по ключам и вид связи
 - Диаграмма
3. Приложение: Скрипты для создания БД
4. Приложение: Скрипты для создания таблиц

Полный набор скриптов в формате .sql находится вместе с отчетом в общем архивном файле индивидуального задания:

create_db.sql	(скрипт для создания Базы Данных)
create_tables.sql	(скрипт для создания таблиц и ограничений)
load_db.sql	(скрипт для заполнения БД значениями)
create_views.sql	(скрипт для создания представлений)
create_triggers.sql	(скрипт для создания триггеров)
create_sp.sql	(скрипт для создания хранимых процедур)
scriptForTesting.sql	(клиентский скрипт для тестирования БД)
SQLServer_script.sql	(скрипт БД, выгруженный из SQL SERVER)

1. Описание системы

Название базы данных: VideoRental

Сущности:

1. Клиенты (таблица Clients)
2. Фильмы (таблица Movies)
3. Кассеты (таблица Tapes)
4. Тарифы (таблица Tariffs)
5. Заявки (таблица Proposals)
6. Аренда (таблица Rental)

Связи между сущностями:

1. Каталог Тарифов - Тариф для кассеты
2. Каталог Фильмов - Фильм на кассете
3. Каталог Кассет - Операция аренды кассеты
4. База Клиентов - Операция аренды для клиента
5. Каталог Фильмов - Заявка на добавление фильма
6. База Клиентов - Заявка на добавление фильма от клиента

Все связи осуществляются внешними ключами (FOREIGN KEY) с использованием ID-колонок таблиц.

Функциональность системы:

Добавление записей:

В таблицу **Clients** добавление новой записи (нового клиента) производит работник видеопроката. Первичный ключ (ClientID) присваивается автоматически. Все атрибуты обязательны к заполнению, при этом атрибут Discount и Balance имеют нулевые значения по умолчанию (Default Constraint).

В таблицу **Tariffs** добавление новой записи (нового тарифа) производит работник видеопроката. Первичный ключ (TariffID) в виде названия указывается вручную. Все атрибуты обязательны к заполнению.

В таблицу **Movies** добавление новой фильма производит работник видеопроката с помощью хранимой процедуры AddTape (параметры: @Title, @Genre, @ReleaseDate, @Duration, @Discription, @Type, @DateManufacture, @TariffID). Первичный ключ (MovieID) присваивается автоматически. Все атрибуты обязательны к заполнению.

В таблицу **Proposals** добавление новой заявки на фильм производит работник видеопроката с помощью хранимой процедуры AddProposal (параметры: @ClientID, @NewMovieTitle). Заявки принимаются только от постоянных клиентов (которые брали кассеты в прокат 5 и более раз).

Первичный ключ (ProposalID) присваивается автоматически. Дата заявки (DateOfProposal) – текущая дата. Атрибуты DateOfAddiction и MovieID допускают пустые значения (они заполняются только когда будет принято решение добавить указанный в заявке фильм в каталог). В таблице два внешних ключа: ClientID (ссылается на РК таблицы Clients со всеми клиентами) и MovieID (ссылается на РК таблицы Movies – каталог с фильмами).

В таблицу **Tapes** добавление новой кассеты и нового фильма производит работник видеопроката с помощью хранимой процедуры: AddTape (параметры: @Title, @Genre, @ReleaseDate, @Duration, @Discription, @Type, @DateManufacture, @TariffID). Обновление производят

на основании таблицы с заявками – Proposals. Первичный ключ (TapeID) присваивается автоматически. Все атрибуты обязательны к заполнению, при этом атрибут Available имеет значение по умолчанию: "1" (этот статус означает, что кассета доступна для аренды, в то время как "0" – недоступна). Таблица имеет два внешних ключа: TariffID (ссылается на РК таблицы Tariffs со всеми тарифами) и MovieID (ссылается на РК таблицы Movies – каталог с фильмами).

В таблицу **Rental** добавление новой операции выдачи кассеты в прокат производит работник видеопроката с помощью хранимой процедуры: RentTape (параметры: @ClientID, @TapeID, @OutDate, @DueDate), которая добавляет запись и автоматически рассчитывает итоговую стоимость проката, в зависимости от дней аренды и соответствующих расценок тарифа (данные по которому хранятся в другой таблице). Первичный ключ (RentalID) присваивается автоматически. Подлежат заполнению все атрибуты за исключением трех: TapeID может быть пустым, в случае, если кассета была повреждена и удалена из каталога; ReturnDate и Fine заполняются после возврата кассеты, поэтому изначально будут пустыми. Атрибут Deposit (залог) имеет два значения: "1" - залог получен и "0" - залог возвращен.

Таблица имеет два внешних ключа: ClientID (ссылается на PK таблицы Clients со всеми клиентами) и TapeID (ссылается на PK таблицы Tapes - каталог с кассетами). Кроме того, при добавлении новой записи в таблицу Rental работают два триггера: TR_Tapes_NotAvailable проставит статус арендованной кассеты в "0" (недоступна) в таблице Tapes; TR_Client_Balance_credit минусует баланс клиента (Balance в таблице Clients), арендовавшего кассету, на стоимость аренды (RentPrice из таблицы Rental). Отрицательный баланс показывает, что у клиента есть задолженность за пользование кассеты.

Связь 1. Каталог Тарифов - Тариф для кассеты. Через две таблицы: Tariffs.TariffID - Tapes.TariffID (PK - FK)

Связь 2. Каталог Фильмов - Фильм на кассете. Через две таблицы: Movies.MovieID - Tapes.MovieID (PK - FK)

Связь 3. Каталог Кассет - Операция аренды кассеты. Через две таблицы: Tapes.TapeID - Rental.TapeID (PK - FK)

Связь 4. База Клиентов - Операция аренды для клиента. Через две таблицы: Clients.ClientID - Rental.ClientID (PK - FK)

Связь 5. Каталог Фильмов - Заявка на добавление фильма. Через две таблицы: Movies.MovieID - Proposals.MovieID (PK - FK)

Связь 6. База Клиентов - Заявка на добавление фильма от клиента. Через две таблицы:
Clients.ClientID - Proposals.ClientID (PK - FK)

Редактирование сущностей:

В таблице **Clients** возможны обновления данных о клиентах с помощью UPDATE TABLE, за исключением атрибута Balance (обновляется через триггер на таблице Rental).

В таблице **Tariffs** возможны обновления в тарифах с помощью UPDATE TABLE.

В таблице **Movies** возможны обновления в информации о фильмах с помощью UPDATE TABLE.

В таблице **Proposals** обновление происходит вместе с вызываемой процедурой AddTape: когда добавляется новый фильм и кассета в каталог видеопроката из списка заявок – в таблице Proposals автоматически обновляется дата добавления фильма и присвоенный ID этому фильму (по соответствующей заявке).

В таблице **Tapes** возможны как обновление так и удаление записей. Обновление происходит в колонке Available каждый раз, когда кассету берут в аренду или возвращают (через триггер на таблице Rental). Удаление происходит, когда кассета была повреждена клиентом во время аренды и подлежит удалению из каталога, при этом залог клиенту не возвращается. Для удаления записи используется хранимая процедура RemoveTape (параметры: @TapeID, @ClientID). Запись из таблицы Tapes полностью удаляется. А в таблице Rental, где существует внешний ключ к этой таблице происходит обновление двух полей: в TapeID проставляется NULL, а в Deposit статус "1" (т.е. залог остался в прокате).

Таблица **Rental** обновляется когда происходит возврат выданной кассеты обратно в видеопрокат с оплатой стоимости аренды. Эту операцию производит хранимая процедура ReturnTape (параметры: @RentalID, @ReturnDate). Обновляются поля Deposit (статус "0" – залог возвращен), ReturnDate (фактической датой возврата) и, если она оказалась позже запланированной (т.е. просрочка аренды), то автоматически начисляется штраф исходя из расценок тарифа. При этом также обновляется поле Balance таблицы Clients: оплаченная стоимость аренды плюсуется в итоговой сумме баланса (обновление через триггер).

Требования к поддержке целостности данных

- Кроме таблицы Tapes записи в таблицах не удаляются, только обновляются (т.к. это история операций проката).
- Если происходит удаление кассеты из таблицы Tapes, то в таблице Rental в соответствующем поле TapeID, которое по внешнему ключу ссылается на удаленную запись, будет проставлен NULL.
- В таблице Clients настроено СНЕК ограничение на требуемый формат номера телефона.
- В таблице Rental настроено СНЕК ограничение на проверку положительности значения стоимости аренды.
- В таблице Rental настроены СНЕК ограничения на проверку, что дата возврата (фактическая и запланированная) не могут быть раньше даты выдачи кассеты.
- В таблице Tariffs настроено СНЕК ограничение на проверку, что стоимость штрафа за просрочку не может быть больше стоимость самой аренды.
- Баланс клиента в таблице Client обновляется только исходя из изменений в таблице Rental (по соответствующим триггерам)

Другие возможности системы

В системе настроены следующие представления (VIEW) :

- Представление MoviesCatalogForClients создано как каталог для клиентов видеопроката. Оно включает в себя колонки 3 таблиц: вся информация о фильме из таблицы Movies, ID и тип кассеты и ее доступность для аренды из таблицы Tapes, а также основная информация о тарифе из таблицы Tariffs.
- Представление CostReportForManagement создано как отчет для менеджеров/бухгалтеров/руководства. Содержит информацию про денежные расчеты, цены и балансы. Включает две таблицы: Rental и Clients.
- Представление ActivityReportForManagement создано как отчет для руководства. Включает топ-5 самых популярных фильма, которые берут в прокате. Выводит название и ID фильма из таблицы Movies и общее количество, сколько этот фильм арендовали из таблицы Rental.

- Представление ProposalReportForManagement создано как отчет для руководства. Выводит информацию о предложениях от постоянных клиентов на пополнение каталога новыми фильмами из таблицы Proposals (без учета тех, которые уже были по просьбам добавлены).

Также в системе реализованы несколько хранимых процедур (STORED PROCEDURE) для быстрого доступа к часто запрашиваемой информации:

- Процедура PlanningReturnDate (параметры: @TapeID) показывает предполагаемую дату возврата определенной кассеты.
- Процедура FindMovie (параметры: @Title) осуществляет поиск по каталогу фильмов. Если фильм в каталоге присутствует будет выведена также информация о кассете и ее доступности для аренды
- Процедура IsRegularClient (параметры: @ClientID) проверяет, является ли клиент постоянным или нет.

2. Структура системы

Таблицы и типы данных в них

Clients			
	Column Name	Data Type	Allow Nulls
🔑	ClientID	int	<input type="checkbox"/>
	LastName	nvarchar(20)	<input type="checkbox"/>
	FirstName	nvarchar(10)	<input type="checkbox"/>
	Address	nvarchar(60)	<input type="checkbox"/>
	Phone_no	char(12)	<input type="checkbox"/>
	Discount	real	<input type="checkbox"/>
	Balance	money	<input type="checkbox"/>
			<input type="checkbox"/>

Movies			
	Column Name	Data Type	Allow Nulls
🔑	MovieID	int	<input type="checkbox"/>
	Title	nvarchar(50)	<input type="checkbox"/>
	Genre	nvarchar(50)	<input type="checkbox"/>
	ReleaseDate	date	<input type="checkbox"/>
	Duration	time(7)	<input type="checkbox"/>
	Discription	nvarchar(250)	<input type="checkbox"/>
			<input type="checkbox"/>

Proposals			
	Column Name	Data Type	Allow Nulls
🔑	ProposalID	int	<input type="checkbox"/>
	ClientID	int	<input type="checkbox"/>
	NewMovieTitle	nvarchar(50)	<input type="checkbox"/>
	DateOfProposal	date	<input type="checkbox"/>
	DateOfAddiction	date	<input checked="" type="checkbox"/>
	MovieID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Rental			
	Column Name	Data Type	Allow Nulls
🔑	RentalID	int	<input type="checkbox"/>
	ClientID	int	<input type="checkbox"/>
	TapelID	int	<input checked="" type="checkbox"/>
	OutDate	date	<input type="checkbox"/>
	DueDate	date	<input type="checkbox"/>
	Deposit	char(1)	<input type="checkbox"/>
	RentPrice	money	<input type="checkbox"/>
	ReturnDate	date	<input checked="" type="checkbox"/>
	Fine	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tapes			
	Column Name	Data Type	Allow Nulls
🔑	TapelID	int	<input type="checkbox"/>
	MovieID	int	<input type="checkbox"/>
	Type	nvarchar(10)	<input type="checkbox"/>
	DateManufacture	date	<input type="checkbox"/>
	TariffID	nvarchar(20)	<input type="checkbox"/>
	Available	char(1)	<input type="checkbox"/>
			<input type="checkbox"/>

Tariffs			
	Column Name	Data Type	Allow Nulls
🔑	TariffID	nvarchar(20)	<input type="checkbox"/>
	PricePerDay	money	<input type="checkbox"/>
	DepositAmount	money	<input type="checkbox"/>
	FinePerDay	money	<input type="checkbox"/>
			<input type="checkbox"/>

Примеры данных для таблиц

Таблица *Clients*:

	ClientID	LastName	FirstName	Address	Phone_no	Discount	Balance
1	1	Pavarotti	Jose	187 Suffolk Ln.	+72085558097	0	0.00
2	2	Kumar	Hari	90 Wadhurst Rd.	+79505551717	0.1	-225.00
3	3	Petersen	Jytte	Vinbaltet 34	+79101234564	0	0.00
4	4	Perrier	Dominique	25, rue Lauriston	+79064755601	0	0.00
5	5	Braunschweiler	Artem	P.O. Box 555	+73075554680	0	0.00

Таблица *Movies*:

	MovieID	Title	Genre	ReleaseDate	Duration	Discription
1	1	Gladiator	history	2000-05-01	02:35:00.0000000	A former Roman General sets out to exact vengeance...
2	2	The Godfather	crime	1972-03-14	02:55:00.0000000	The Godfather follows Vito Corleone, Don of the Cor...
3	3	The Shawshank Redemption	drama	1994-08-27	02:22:00.0000000	Two imprisoned men bond over a number of years, fi...
4	4	Schindlers List	history	1993-11-30	03:15:00.0000000	In German-occupied Poland during World War II, in...
5	5	Pulp Fiction	crime	1994-05-21	02:34:00.0000000	The lives of two mob hitmen, a boxer, a gangster an...

Таблица *Tapes*:

	TapeID	MovieID	Type	DateManufacture	TariffID	Available
1	1	1	VHS	2000-12-01	Tariff2	1
2	2	2	VHS	1999-11-05	Tariff1	1
3	3	3	VHS	1995-12-10	Tariff1	1
4	4	4	VHS	1998-06-01	Tariff2	1
5	5	5	VHS	1999-12-14	Tariff1	0

Таблица *Tariffs*:

	TariffID	PricePerDay	DepositAmount	FinePerDay
1	Tariff1	50.00	1000.00	5.00
2	Tariff2	100.00	2000.00	10.00

Таблица *Proposals*:

	ProposalID	ClientID	NewMovieTitle	DateOfProposal	DateOfAddiction	MovieID
1	1	2	Green Book	2021-01-20	2021-02-27	30
2	2	2	Tenet	2021-04-29	NULL	NULL

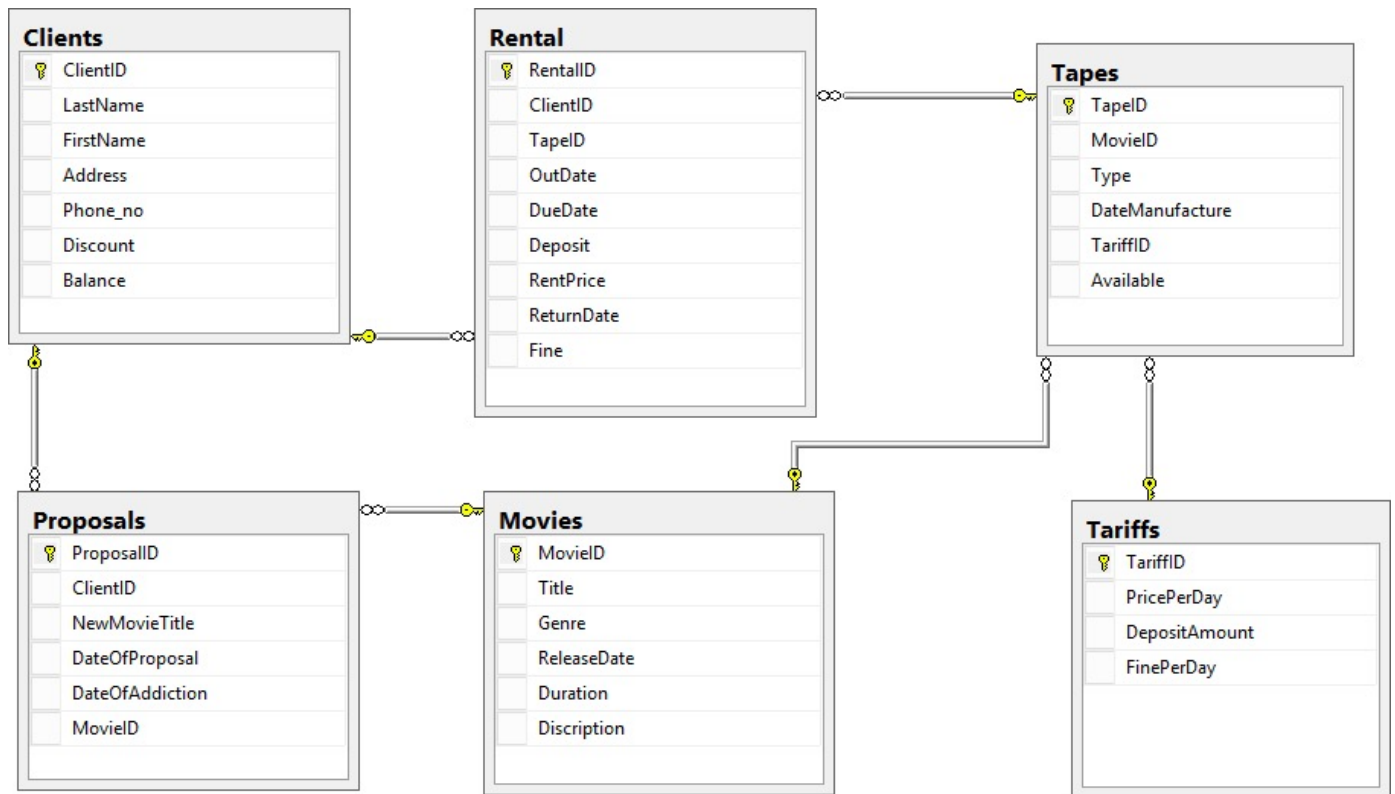
Таблица *Rental*:

	RentalID	ClientID	TapeID	OutDate	DueDate	Deposit	RentPrice	ReturnDate	Fine
1	1	1	37	2021-01-15	2021-01-20	0	500.00	2021-01-20	0.00
2	2	2	17	2021-01-18	2021-01-29	0	1100.00	2021-01-29	0.00
3	3	3	25	2021-02-01	2021-02-04	0	150.00	2021-02-05	5.00
4	4	2	20	2021-02-16	2021-02-18	0	200.00	2021-02-18	0.00
5	5	4	3	2021-03-20	2021-03-27	0	350.00	2021-03-27	0.00

Связи таблиц по ключам и вид связи (1-1, 1-многим)

Таблица 1	Ключ первичный	Таблица 2	Ключ внешний	Мощность связи
Tariffs	TariffID	Tapes	TariffID	1-много
Movies	MovieID	Tapes	MovieID	1-много
Movies	MovieID	Proposal	MovieID	1-1
Clients	ClientID	Proposal	ClientID	1-много
Tapes	TapeID	Rental	TapeID	1-много
Clients	ClientID	Rental	ClientID	1-много

ДИАГРАММА



СВЯЗИ

	Column Name	Data Type	Allow Nulls
PK	ClientID	int	<input type="checkbox"/>
	LastName	nvarchar(20)	<input type="checkbox"/>
	FirstName	nvarchar(10)	<input type="checkbox"/>
	Address	nvarchar(60)	<input type="checkbox"/>
	Phone_no	char(12)	<input type="checkbox"/>
	Discount	real	<input type="checkbox"/>
	Balance	money	<input type="checkbox"/>

Foreign Key Relationships

Selected Relationship:
FK_Proposals_Clients
FK_Rental_Clients

Editing properties for existing relationship.

- (General)**
 - Check Existing Data On Creation Or Re-Enabling: No
- Tables And Columns Specification**

Foreign Key Base Table	Foreign Key Columns	Primary/Unique Key Base Table	Primary/Unique Key Columns
Proposals	ClientID	Clients	ClientID
- Database Designer**
- Identity**

Buttons: Add, Delete, Close

Foreign Key Relationships

Selected Relationship:

FK_Rental_Clients
FK_Rental_Tapes

Editing properties for existing relationship.

(General)

Check Existing Data On Creati No

Tables And Columns Specific

Foreign Key Base Table Rental

Foreign Key Columns ClientID

Primary/Unique Key Base Clients

Primary/Unique Key Colu ClientID

Database Designer

Identity

Add Delete Close

	Column Name	Data Type	Allow Nulls
	RentalID	int	<input type="checkbox"/>
	ClientID	int	<input type="checkbox"/>
	TapelD	int	<input checked="" type="checkbox"/>
	OutDate	date	<input type="checkbox"/>
	DueDate	date	<input type="checkbox"/>
	Deposit	char(1)	<input type="checkbox"/>
	RentPrice	money	<input type="checkbox"/>
	ReturnDate	date	<input checked="" type="checkbox"/>
	Fine	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Foreign Key Relationships

Selected Relationship:

FK_Proposals_Movies
FK_Tapes_Movies

Editing properties for existing relationship.

(General)

Check Existing Data On Creati No

Tables And Columns Specific

Foreign Key Base Table Proposals

Foreign Key Columns MovieID

Primary/Unique Key Base Movies

Primary/Unique Key Colu MovieID

Database Designer

Identity

Add Delete Close

	date	<input checked="" type="checkbox"/>
	money	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
	MovieID	int	<input type="checkbox"/>
	Title	nvarchar(50)	<input type="checkbox"/>
	Genre	nvarchar(50)	<input type="checkbox"/>
	ReleaseDate	date	<input type="checkbox"/>
	Duration	time(7)	<input type="checkbox"/>
	Discription	nvarchar(250)	<input type="checkbox"/>
			<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
--	-------------	-----------	-------------

Foreign Key Relationships

Selected Relationship:

FK_Tapes_Tariffs

Editing properties for existing relationship.

(General)

Check Existing Data On Creati No

Tables And Columns Specific

Foreign Key Base Table Tapes

Foreign Key Columns TariffID

Primary/Unique Key Base Tariffs

Primary/Unique Key Colu TariffID

Database Designer

Identity

Add Delete Close

			<input type="checkbox"/>
--	--	--	--------------------------

	Column Name	Data Type	Allow Nulls
	TariffID	nvarchar(20)	<input type="checkbox"/>
	PricePerDay	money	<input type="checkbox"/>
	DepositAmount	money	<input type="checkbox"/>
	FinePerDay	money	<input type="checkbox"/>
			<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
ProposalID	int	<input type="checkbox"/>
ClientID	int	<input type="checkbox"/>
NewMovieTitle	nvarchar(50)	<input type="checkbox"/>
DateOfProposal	date	<input type="checkbox"/>
DateOfAddiction	date	<input checked="" type="checkbox"/>
MovieID	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
TapelD	int	<input type="checkbox"/>

Foreign Key Relationships

Selected Relationship:
FK_Proposals_Clients
FK_Proposals_Movies

Editing properties for existing relationship.

(General)

Check Existing Data On Creati No

Tables And Columns Specific

Foreign Key Base Table: Proposals

Foreign Key Columns: ClientID

Primary/Unique Key Base: Clients

Primary/Unique Key Colu: ClientID

Database Designer

Identity

AddDeleteClose

Column Name	Data Type	Allow Nulls
TapelD	int	<input type="checkbox"/>
MovieID	int	<input type="checkbox"/>
Type	nvarchar(10)	<input type="checkbox"/>
DateManufacture	date	<input type="checkbox"/>
TariffID	nvarchar(20)	<input type="checkbox"/>
Available	char(1)	<input type="checkbox"/>

Foreign Key Relationships

Selected Relationship:
FK_Rental_Tapes
FK_Tapes_Movies
FK_Tapes_Tariffs

Editing properties for existing relationship.

(General)

Check Existing Data On Creati No

Tables And Columns Specific

Foreign Key Base Table: Rental

Foreign Key Columns: TapelD

Primary/Unique Key Base: Tapes

Primary/Unique Key Colu: TapelD

Database Designer

Identity

AddDeleteClose

3. Приложение: Скрипты для создания БД

```
USE [master]
GO

CREATE DATABASE [VideoRental]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'VideoRental', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\VideoRental.mdf' , SIZE = 50MB ,
MAXSIZE = 70MB, FILEGROWTH = 10% )
    LOG ON
    ( NAME = N'VideoRental_log', FILENAME = N'C:\Program Files\Microsoft
SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\VideoRental_log.ldf' , SIZE =
15MB , MAXSIZE = 20MB , FILEGROWTH = 10%)
GO

ALTER DATABASE [VideoRental] SET COMPATIBILITY_LEVEL = 120
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [VideoRental].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [VideoRental] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [VideoRental] SET ANSI_NULLS OFF
GO
ALTER DATABASE [VideoRental] SET ANSI_PADDING OFF
GO
ALTER DATABASE [VideoRental] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [VideoRental] SET ARITHABORT OFF
GO
ALTER DATABASE [VideoRental] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [VideoRental] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [VideoRental] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [VideoRental] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [VideoRental] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [VideoRental] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [VideoRental] SET NUMERIC_ROUNDABORT OFF
GO
```

```
ALTER DATABASE [VideoRental] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [VideoRental] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [VideoRental] SET DISABLE_BROKER
GO
ALTER DATABASE [VideoRental] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [VideoRental] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [VideoRental] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [VideoRental] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [VideoRental] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [VideoRental] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [VideoRental] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [VideoRental] SET RECOVERY FULL
GO
ALTER DATABASE [VideoRental] SET MULTI_USER
GO
ALTER DATABASE [VideoRental] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [VideoRental] SET DB_CHAINING OFF
GO
ALTER DATABASE [VideoRental] SET FILESTREAM( NON_TRANSACTED_ACCESS =
OFF )
GO
ALTER DATABASE [VideoRental] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
ALTER DATABASE [VideoRental] SET DELAYED_DURABILITY = DISABLED
GO
EXEC sys.sp_db_vardecimal_storage_format N'VideoRental', N'ON'
GO
```

4. Приложение: Скрипты для создания таблиц

```
CREATE TABLE Clients
```

```
(
    ClientID      int IDENTITY (1, 1) NOT NULL
,   LastName     nvarchar (20)      NOT NULL
,   FirstName    nvarchar (10)      NOT NULL
,   Address      nvarchar (60)      NOT NULL
,   Phone_no     char(12)           NOT NULL
,   Discount     real               NOT NULL
,   Balance      money              NOT NULL
)
ON [PRIMARY]
GO
```

```
CREATE TABLE Tapes
```

```
(
    TapeID        int IDENTITY (1, 1) NOT NULL
,   MovieID      int                NOT NULL
,   Type         nvarchar(10)       NOT NULL
,   DateManufacture date            NOT NULL
,   TariffID     nvarchar(20)       NOT NULL
,   Available    char(1)            NOT NULL
)
ON [PRIMARY]
GO
```

```
CREATE TABLE Movies
```

```
(
    MovieID      int IDENTITY (1, 1) NOT NULL
,   Title       nvarchar(50)        NOT NULL
,   Genre       nvarchar(50)        NOT NULL
,   ReleaseDate date                NOT NULL
,   Duration    time                NOT NULL
,   Discription nvarchar(250)       NOT NULL
)
ON [PRIMARY]
GO
```



```

CREATE TABLE Tariffs
(
    TariffID      nvarchar(20)          NOT NULL
,   PricePerDay  money                  NOT NULL
,   DepositAmount money                 NOT NULL
,   FinePerDay   money                  NOT NULL
)
ON [PRIMARY]
GO

```

```

CREATE TABLE Proposals
(
    ProposalID    int IDENTITY (1, 1) NOT NULL
,   ClientID     int                  NOT NULL
,   NewMovieTitle nvarchar(50)        NOT NULL
,   DateOfProposal date               NOT NULL
,   DateOfAddiction date              NULL
,   MovieID      int                 NULL
)
ON [PRIMARY]
GO

```

```

CREATE TABLE Rental
(
    RentalID      int IDENTITY (1, 1) NOT NULL
,   ClientID     int                  NOT NULL
,   TapeID       int                  NULL
,   OutDate      date                 NOT NULL
,   DueDate      date                 NOT NULL
,   Deposit      char(1)              NOT NULL
,   RentPrice    money                 NOT NULL
,   ReturnDate   date                 NULL
,   Fine         money                 NULL
)
ON [PRIMARY]
GO

```

```
ALTER TABLE Clients
    ADD CONSTRAINT PK_Clients PRIMARY KEY CLUSTERED (ClientID)
```

```
ALTER TABLE Rental
    ADD CONSTRAINT PK_Rental PRIMARY KEY CLUSTERED (RentalID)
```

```
ALTER TABLE Tapes
    ADD CONSTRAINT PK_Tapes PRIMARY KEY CLUSTERED (TapeID)
```

```
ALTER TABLE Movies
    ADD CONSTRAINT PK_Movies PRIMARY KEY CLUSTERED (MovieID)
```

```
ALTER TABLE Tariffs
    ADD CONSTRAINT PK_Tariffs PRIMARY KEY CLUSTERED (TariffID)
```

```
ALTER TABLE Proposals
    ADD CONSTRAINT PK_Proposals PRIMARY KEY CLUSTERED (ProposalID)
GO
```

```
ALTER TABLE Rental WITH NOCHECK
    ADD CONSTRAINT FK_Rental_Clients FOREIGN KEY (ClientID) REFERENCES
Clients (ClientID)
```

```
ALTER TABLE Rental WITH NOCHECK
    ADD CONSTRAINT FK_Rental_Tapes FOREIGN KEY (TapeID) REFERENCES
Tapes (TapeID) ON DELETE SET NULL
```

```
ALTER TABLE Tapes WITH NOCHECK
    ADD CONSTRAINT FK_Tapes_Movies FOREIGN KEY (MovieID) REFERENCES
Movies (MovieID)
```

```
ALTER TABLE Tapes WITH NOCHECK
    ADD CONSTRAINT FK_Tapes_Tariffs FOREIGN KEY (TariffID) REFERENCES
Tariffs (TariffID)
```

```
ALTER TABLE Proposals WITH NOCHECK
    ADD CONSTRAINT FK_Proposals_Movies FOREIGN KEY (MovieID) REFERENCES
Movies (MovieID)
```

```
ALTER TABLE Proposals WITH NOCHECK
    ADD CONSTRAINT FK_Proposals_Clients FOREIGN KEY (ClientID)
REFERENCES Clients (ClientID)
GO
```

```
ALTER TABLE Clients
    ADD CONSTRAINT DF_Clients_Discount DEFAULT 0 FOR Discount
```

```
ALTER TABLE Clients
    ADD CONSTRAINT DF_Clients_Balance DEFAULT 0 FOR Balance
```

```
ALTER TABLE Tapes
    ADD CONSTRAINT DF_Tapes DEFAULT '1' FOR Available
GO
```

```
ALTER TABLE Clients WITH NOCHECK
    ADD CONSTRAINT CK_phone CHECK (Phone_no LIKE
'+7[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
```

```
ALTER TABLE Rental WITH NOCHECK
    ADD CONSTRAINT CK_Rental_DueDate CHECK (DueDate > OutDate)
```

```
ALTER TABLE Rental WITH NOCHECK
    ADD CONSTRAINT CK_Rental_ReturnDate CHECK (ReturnDate > OutDate)
```

```
ALTER TABLE Rental WITH NOCHECK
    ADD CONSTRAINT CK_Rental_Price CHECK (RentPrice > 0)
```

```
ALTER TABLE Tariffs WITH NOCHECK
    ADD CONSTRAINT CK_Tariffs CHECK (FinePerDay < PricePerDay)
GO
```

Полный набор скриптов в формате .sql находится вместе с отчетом в общем архивном файле индивидуального задания. В том числе файл **scriptForTesting.sql**. В нем подробно описываются все возможности системы и демонстрируется ее работа на конкретных примерах.