НАО «КАРАГАНДИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМЕНИ АБЫЛКАСА САГИНОВА»

Кафедра информационных технологий и безопасности

**Лабораторная работа № 3**

**Дисциплина:** "Введение в базы данных"

**Тема:** Создание запросов и модификация таблиц базы данных

**Принял:**

Преподаватель: Жакина М.М

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)      (дата)*

**Выполнил:**

ст.гр.СИБ-23-7  Аксеной А.И

Караганда 2025

**Лабораторная работа №3**

**Тема:** Создание запросов и модификация таблиц базы данных  
  
**Цель:** Усвоить способы создания выборки данных в среде СУБД MS SQL Server.

**Индивидуальное задание:**

|  |
| --- |
| 1. На каких дисках находится «*i-ый*» фильм 2. Все фильмы в названии которых есть слово «. . .» 3. Список клиентов, задержавших диск  на текущую дату 4. Общее количество фильмов, имеющихся в фильмотеке каждого из режиссеров   Какие жанры фильмов  интересовали  пользователей в текущем месяце |

**Ход работы:**

CREATE DATABASE laba3

ON (

NAME = 'laba3',

FILENAME = 'C:\MSSQL\laba3.mdf',

SIZE = 10MB,

MAXSIZE = 100MB,

FILEGROWTH = 5MB

)

LOG ON (

NAME = 'laba3\_log',

FILENAME = 'C:\MSSQL\laba3.ldf',

SIZE = 20MB,

MAXSIZE = 100MB,

FILEGROWTH = 5MB

);

GO

USE laba3;

GO

CREATE TABLE Genres (

GenreID INT IDENTITY(1,1) CONSTRAINT PK\_Genres PRIMARY KEY,

GenreName NVARCHAR(100) NOT NULL CONSTRAINT UQ\_Genres\_GenreName UNIQUE

);

GO

CREATE UNIQUE INDEX IX\_Genres\_GenreName ON Genres(GenreName);

GO

CREATE TABLE Countries (

CountryID INT IDENTITY(1,1) CONSTRAINT PK\_Countries PRIMARY KEY,

CountryName NVARCHAR(100) NOT NULL CONSTRAINT UQ\_Countries\_CountryName UNIQUE

);

GO

CREATE UNIQUE INDEX IX\_Countries\_CountryName ON Countries(CountryName);

GO

CREATE TABLE Directors (

DirectorID INT IDENTITY(1,1) CONSTRAINT PK\_Directors PRIMARY KEY,

FullName NVARCHAR(150) NOT NULL

);

GO

CREATE TABLE DepositTypes (

DepositTypeID INT IDENTITY(1,1) CONSTRAINT PK\_DepositTypes PRIMARY KEY,

DepositName NVARCHAR(100) NOT NULL CONSTRAINT UQ\_DepositTypes\_DepositName UNIQUE

);

GO

CREATE UNIQUE INDEX IX\_DepositTypes\_DepositName ON DepositTypes(DepositName);

GO

CREATE TABLE Films (

FilmID INT IDENTITY(1,1) CONSTRAINT PK\_Films PRIMARY KEY,

Title NVARCHAR(200) NOT NULL,

ReleaseDate DATE NOT NULL CONSTRAINT CHK\_Films\_ReleaseDate CHECK (ReleaseDate >= '1900-01-01'),

GenreID INT NOT NULL,

DirectorID INT NOT NULL,

CountryID INT NOT NULL,

Duration FLOAT NOT NULL CONSTRAINT CHK\_Films\_Duration CHECK (Duration > 0),

Description NVARCHAR(MAX) DEFAULT N'Описание отсутствует',

CONSTRAINT FK\_Films\_Genre FOREIGN KEY (GenreID)

REFERENCES Genres(GenreID)

ON DELETE NO ACTION

ON UPDATE CASCADE,

CONSTRAINT FK\_Films\_Director FOREIGN KEY (DirectorID)

REFERENCES Directors(DirectorID)

ON DELETE NO ACTION

ON UPDATE CASCADE,

CONSTRAINT FK\_Films\_Country FOREIGN KEY (CountryID)

REFERENCES Countries(CountryID)

ON DELETE NO ACTION

ON UPDATE CASCADE

);

GO

CREATE TABLE Disks (

DiskID INT IDENTITY(1,1) CONSTRAINT PK\_Disks PRIMARY KEY,

DepositPrice DECIMAL(10,2) NOT NULL CONSTRAINT CHK\_Disks\_DepositPrice CHECK (DepositPrice >= 0),

RentPricePerDay DECIMAL(10,2) NOT NULL CONSTRAINT CHK\_Disks\_RentPrice CHECK (RentPricePerDay >= 0),

TotalQuantity INT NOT NULL CONSTRAINT CHK\_Disks\_TotalQuantity CHECK (TotalQuantity > 0),

AvailableQuantity INT NOT NULL,

DateAdded DATE NOT NULL DEFAULT GETDATE(),

CONSTRAINT CHK\_Disks\_Availability CHECK (

AvailableQuantity >= 0

AND AvailableQuantity <= TotalQuantity

)

);

GO

CREATE TABLE DiskContent (

DiskID INT NOT NULL,

FilmID INT NOT NULL,

CONSTRAINT PK\_DiskContent PRIMARY KEY (DiskID, FilmID),

CONSTRAINT FK\_DiskContent\_Disk FOREIGN KEY (DiskID)

REFERENCES Disks(DiskID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_DiskContent\_Film FOREIGN KEY (FilmID)

REFERENCES Films(FilmID)

ON DELETE CASCADE

ON UPDATE CASCADE

);

GO

CREATE TABLE Clients (

ClientID INT IDENTITY(1,1) CONSTRAINT PK\_Clients PRIMARY KEY,

FullName NVARCHAR(150) NOT NULL,

Address NVARCHAR(200) DEFAULT N'Не указано',

Phone NVARCHAR(20) NOT NULL CHECK (Phone LIKE '[0-9+ -]%')

);

GO

CREATE TABLE Movements (

MovementID INT IDENTITY(1,1) CONSTRAINT PK\_Movements PRIMARY KEY,

DiskID INT NOT NULL,

ClientID INT NOT NULL,

IssueDate DATE NOT NULL DEFAULT GETDATE(),

RentPeriod INT NOT NULL CONSTRAINT CHK\_Movements\_RentPeriod CHECK (RentPeriod > 0),

DepositTypeID INT NOT NULL,

ReturnDate DATE,

CONSTRAINT CHK\_Movements\_ReturnDate CHECK (

ReturnDate IS NULL

OR ReturnDate >= IssueDate

),

CONSTRAINT FK\_Movements\_Disk FOREIGN KEY (DiskID)

REFERENCES Disks(DiskID)

ON DELETE NO ACTION

ON UPDATE CASCADE,

CONSTRAINT FK\_Movements\_Client FOREIGN KEY (ClientID)

REFERENCES Clients(ClientID)

ON DELETE NO ACTION

ON UPDATE CASCADE,

CONSTRAINT FK\_Movements\_DepositType FOREIGN KEY (DepositTypeID)

REFERENCES DepositTypes(DepositTypeID)

ON DELETE NO ACTION

ON UPDATE CASCADE

);

GO

CREATE TABLE Payments (

PaymentID INT IDENTITY(1,1) CONSTRAINT PK\_Payments PRIMARY KEY,

ClientID INT NOT NULL,

PaymentDate DATE NOT NULL DEFAULT GETDATE(),

Amount DECIMAL(10,2) NOT NULL CONSTRAINT CHK\_Payments\_Amount CHECK (Amount > 0),

CONSTRAINT FK\_Payments\_Client FOREIGN KEY (ClientID)

REFERENCES Clients(ClientID)

ON DELETE NO ACTION

ON UPDATE CASCADE

);

GO

**Примерные данные:**

INSERT INTO Genres (GenreName) VALUES (N'Драма'), (N'Комедия'), (N'Боевик'), (N'Фантастика');

INSERT INTO Countries (CountryName) VALUES (N'США'), (N'Франция'), (N'Япония');

INSERT INTO Directors (FullName) VALUES (N'Кристофер Нолан'), (N'Квентин Тарантино');

INSERT INTO Films (Title, ReleaseDate, GenreID, DirectorID, CountryID, Duration)

VALUES

(N'Начало', '2010-07-16', 4, 1, 1, 148),

(N'Джанго освобожденный', '2012-12-25', 3, 2, 1, 165),

(N'Интерстеллар', '2014-11-07', 4, 1, 1, 169),

(N'Криминальное чтиво', '1994-10-14', 1, 2, 1, 154),

(N'Комедийная драма', '2020-06-20', 2, 2, 2, 120);

INSERT INTO Disks (DepositPrice, RentPricePerDay, TotalQuantity, AvailableQuantity)

VALUES (200, 50, 5, 3), (150, 40, 4, 1), (180, 45, 3, 0);

INSERT INTO DiskContent (DiskID, FilmID) VALUES (1, 1), (1, 3), (2, 2), (2, 4), (3, 5);

INSERT INTO Clients (FullName, Address, Phone)

VALUES

(N'Иван Иванов', N'Москва, ул. Пушкина, 10', N'+79991234567'),

(N'Мария Петрова', N'Санкт-Петербург, Невский пр., 25', N'+79997654321');

INSERT INTO DepositTypes (DepositName) VALUES (N'Паспорт'), (N'Деньги');

INSERT INTO Movements (DiskID, ClientID, IssueDate, RentPeriod, DepositTypeID)

VALUES

(1, 1, DATEADD(DAY, -5, GETDATE()), 3, 1),

(2, 2, DATEADD(DAY, -2, GETDATE()), 5, 2);

**Проверка по примерным данным:**

**1) На каких дисках находится фильм "Начало":**

SELECT D.DiskID, D.TotalQuantity, D.AvailableQuantity

FROM Films F

JOIN DiskContent DC ON F.FilmID = DC.FilmID

JOIN Disks D ON DC.DiskID = D.DiskID

WHERE F.Title = N'Начало';

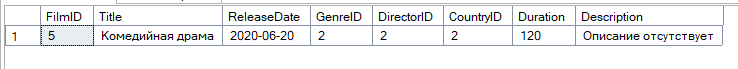


**2) Все фильмы, в названии которых есть слово "драма":**

SELECT \*

FROM Films

WHERE Title LIKE N'%драма%';



**3) Список клиентов, задержавших диск на текущую дату:**

SELECT

C.FullName,

M.DiskID,

M.IssueDate,

M.RentPeriod,

DATEADD(DAY, M.RentPeriod, M.IssueDate) AS DueDate

FROM Movements M

JOIN Clients C ON M.ClientID = C.ClientID

WHERE M.ReturnDate IS NULL

AND DATEADD(DAY, M.RentPeriod, M.IssueDate) < CAST(GETDATE() AS DATE);



**4) Общее количество фильмов у каждого режиссера:**

SELECT

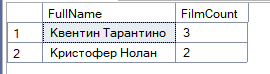
D.FullName,

COUNT(F.FilmID) AS FilmCount

FROM Directors D

LEFT JOIN Films F ON D.DirectorID = F.DirectorID

GROUP BY D.FullName;



**5) Какие жанры интересовали пользователей в текущем месяце:**

SELECT

G.GenreName,

COUNT(\*) AS IssueCount

FROM Movements M

JOIN DiskContent DC ON M.DiskID = DC.DiskID

JOIN Films F ON DC.FilmID = F.FilmID

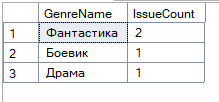
JOIN Genres G ON F.GenreID = G.GenreID

WHERE MONTH(M.IssueDate) = MONTH(GETDATE())

AND YEAR(M.IssueDate) = YEAR(GETDATE())

GROUP BY G.GenreName

ORDER BY IssueCount DESC;



**Контрольные вопросы**:

**1. Возможности меню Query Editor**

Представления множества функций для написания, запуска, отладки и форматирования SQL-запросов.

**Создание и редактирование запросов:**

SQL-запросы по типу: SELECT, INSERT, UPDATE, DELETE, CREATE, работа с несколькими вкладками, подсветка синтаксиса, слов и функций, автозаполнение SQL-кода(IntelliSense).

**Выполнение запросов:**

F5 / Ctrl + E– выполнение текущего запроса  
Ctrl + F5 – выполнение без получения плана выполнения.

**Анализ и отладка:**

Ctrl + M – показ плана выполнения   
Client Statistics – показ статистики выполнения (время, байты и т.д)  
SQL Profiler – для отслеживания запросов

**Инструменты и доп.ф-ции:**

IntelliSense(Ctrl + Space) – автозаполнение  
Свертывание кода  
Подсветка ошибок в реальном времени  
Сохранение и открытие скриптов (.sql файлы)

**Навигация и поиск:**

Поиск и замена (Ctrl + F / Ctrl + H)  
Быстрый переход к объекту, таблице, представлению и т.п  
Перепрыгивание между BEGIN...END, IF...ELSE, TRY...CATCH блоками

**Работа с подключениями:**

Изменение БД контекста(выпадающий список)  
Переключение между серверами  
Подключение с другими учетными данными

**2. Опишите конструкцию оператора SELECT**

Основной инструмент для извлечения данных из базы данных. Позволяет выбрать нужные столбцы и строки из одной или нескольких таблиц с возможностью фильтрации, сортировки, объединения и группировки данных.   
  
**Самый простой запрос:**

SELECT FullName, Phone  
FROM Clients;

**С фильтрацией:**

SELECT \*

FROM Films

WHERE GenreID = 1 AND ReleaseDate >= '2000-01-01';

**С группировкой:**

SELECT DirectorID, COUNT(\*) AS FilmCount

FROM Films

GROUP BY DirectorID

HAVING COUNT(\*) > 1;

**С сортировкой:**

SELECT Title, ReleaseDate

FROM Films

ORDER BY ReleaseDate DESC;

**С объединением таблиц:**

SELECT F.Title, G.GenreName

FROM Films F

JOIN Genres G ON F.GenreID = G.GenreID;