НАО «КАРАГАНДИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМЕНИ АБЫЛКАСА САГИНОВА»

Кафедра информационных технологий и безопасности

**Лабораторная работа № 4**

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

**Тема:** Запросы на выборку данных по вариантам индивидуальных заданий

**Принял:**

Преподаватель: Жакина М.М

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

*(подпись)      (дата)*

**Выполнил:**

ст.гр.СИБ-23-7  Аксеной А.И

Караганда 2025

**Лабораторная работа №** **4**

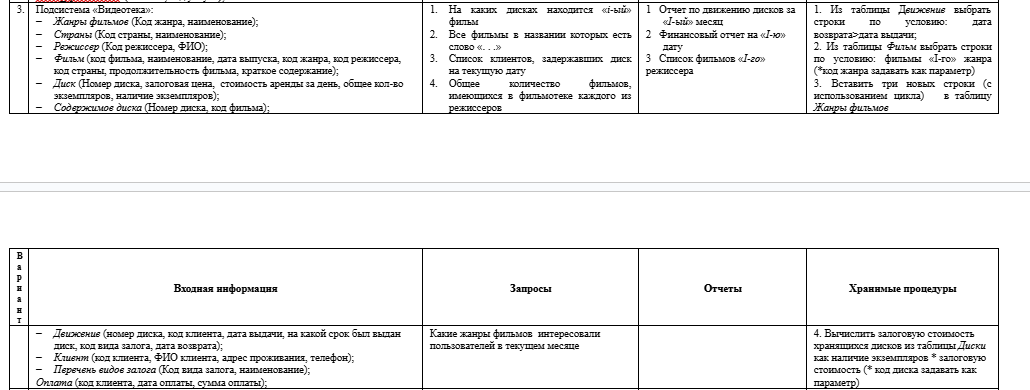
**Тема:** Запросы на выборку данных по вариантам индивидуальных заданий.

**Цель:** Усвоить способы создания представлений средствами СУБД MS SQL Server.

**Индивидуальное задание:**

1. Создать представления для реализации отчетов в соответствии с требованиями индивидуального варианта задания на разработку курсового проекта
2. Создать представление по своему усмотрению на основании двух или более таблиц вашей базы данных с целью просмотра и корректировки базовых таблиц
3. Создать представление по своему усмотрению на основании одной таблицы вашей базы данных с целью корректировки базовой таблицы

**Требования ИД на разработку курсового проекта:**



**Ход работы:**

**ИД 1:** На каких дисках находится i-ый фильм

CREATE VIEW VW\_FilmDiscs AS

SELECT

f.FilmName,

d.DiscID,

d.AvailableCount,

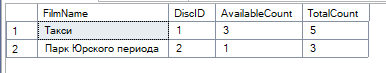
d.TotalCount

FROM Films f

JOIN DiscContent dc ON f.FilmID = dc.FilmID

JOIN Discs d ON dc.DiscID = d.DiscID;

GO



**Поиск по слову:**

SELECT \* FROM VW\_FilmDiscs WHERE FilmName = N'Такси';



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

CREATE VIEW VW\_OverdueClients AS

SELECT

c.ClientID,

c.FullName AS ClientName,

m.DiscID,

DATEADD(DAY, m.DaysIssued, m.IssueDate) AS ExpectedReturnDate,

m.ReturnDate,

DATEDIFF(DAY, DATEADD(DAY, m.DaysIssued, m.IssueDate), GETDATE()) AS DaysOverdue

FROM Movements m

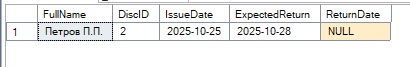
JOIN Clients c ON m.ClientID = c.ClientID

WHERE m.ReturnDate IS NULL

AND DATEADD(DAY, m.DaysIssued, m.IssueDate) < GETDATE();

GO

SELECT \* FROM VW\_OverdueClients;



**Сколько снял каждый режиссер:**

CREATE VIEW VW\_FilmsByDirector AS

SELECT

d.DirectorID,

d.FullName AS DirectorName,

COUNT(f.FilmID) AS TotalFilms

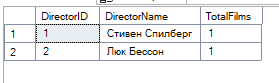
FROM Directors d

LEFT JOIN Films f ON d.DirectorID = f.DirectorID

GROUP BY d.DirectorID, d.FullName;

GO

SELECT \* FROM VW\_FilmsByDirector;



**Интересующие жанры в текущем месяце:**

CREATE VIEW VW\_PopularGenresThisMonth AS

SELECT

g.GenreName,

COUNT(DISTINCT m.MovementID) AS RentalsCount

FROM Movements m

JOIN DiscContent dc ON m.DiscID = dc.DiscID

JOIN Films f ON dc.FilmID = f.FilmID

JOIN Genres g ON f.GenreID = g.GenreID

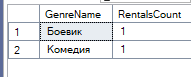
WHERE MONTH(m.IssueDate) = MONTH(GETDATE())

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

GROUP BY g.GenreName;

GO

SELECT \* FROM VW\_PopularGenresThisMonth;



**ИД 2:**

CREATE VIEW VW\_ClientFilmRentals

AS

SELECT

m.MovementID,

c.ClientID,

c.FullName AS ClientName,

f.FilmID,

f.FilmName,

m.DiscID,

m.IssueDate,

m.DaysIssued,

m.ReturnDate,

m.PledgeTypeID

FROM Movements m

JOIN Clients c ON m.ClientID = c.ClientID

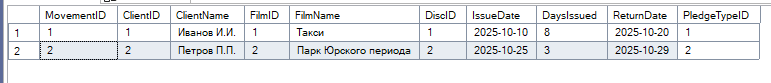
JOIN DiscContent dc ON m.DiscID = dc.DiscID

JOIN Films f ON dc.FilmID = f.FilmID;

GO

**Проверяем:**

SELECT \* FROM VW\_ClientFilmRentals;



-- Допустим, мы захотим изменить дату возврата для нужной выдачи

UPDATE VW\_ClientFilmRentals

SET ReturnDate = '2025-10-29'

WHERE MovementID = 2;

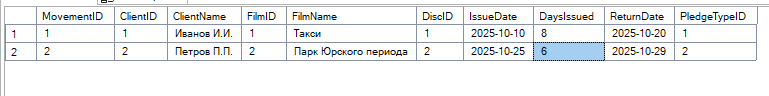
-- Либо можем продлить срок выдачи

UPDATE VW\_ClientFilmRentals

SET DaysIssued = DaysIssued + 3

WHERE MovementID = 2;

**Стало:**



**ИД3: Контроль количества и стоимости дисков**

CREATE VIEW VW\_DiscsStatus

AS

SELECT

DiscID,

DepositPrice,

RentPrice,

TotalCount,

AvailableCount,

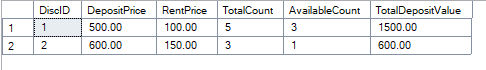
(DepositPrice \* AvailableCount) AS TotalDepositValue

FROM Discs;

GO

**Проверяем:**

SELECT \* FROM VW\_DiscsStatus;



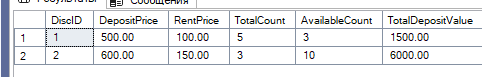
-- Добавим диску с ID = 2, что он теперь имеет количество 10шт

UPDATE VW\_DiscsStatus

SET AvailableCount = 10

WHERE DiscID = 2;

**Стало:**



**Контрольные вопросы**:

1. **Назначение механизма представлений**

VIEW – это VR таблица, которую создаем на основе одного или нескольких запросов к базовым таблицам. Данные физически не хранятся, а хранится только SQL-запрос.   
Обычно используют для упрощения работы с данными, сокрытия сложности структуры базы(связи ключей не показывают пользователю, пользователь имеет доступ только нужные поля через VIEW и соответственно мы хорошо защищаем нашу БД), создаем vr-отчеты, обеспечиваем удобство обновлений INSERT, UPDATE, DELETE.

1. **Классификация представлений**

**Способ создания:**

Статика(постоянные): CREATE VIEW VW\_Clients AS SELECT FullName, Phone FROM Clients;

Динамические(временные): SELECT \* FROM (SELECT \* FROM Films WHERE GenreID = 1) AS temp;

**Структура:**

Простые представления(основа на одной таблице, и можно редактировать через INSERT, UPDATE, DELETE): CREATE VIEW VW\_DiscsStatus AS SELECT DiscID, AvailableCount FROM Discs;

Сложные представления(основа на 2 и более таблицах, есть соединения по типу JOIN):

CREATE VIEW VW\_DiscsStatus

AS

SELECT

DiscID,

DepositPrice,

RentPrice,

TotalCount,

AvailableCount,

(DepositPrice \* AvailableCount) AS TotalDepositValue

FROM Discs;

GO

**Назначение:**

**Представления для отчетов:** Используют для формирования аналитических и статических данных.

**Представления для корректировки данных:** Позволяют редактировать содержимое базовых таблиц.

**Представления для ограничения доступа:** Отображение только части данных, с сокрытием конфиденциальной информации.

**Возможности обновления:**

Обновляемые: Можно выполнять(INSERT, UPDATE, DELETE), если представление основано на одной таблице и не содержит(DISTINCT, GROUP BY, UNION).

Не обновляемые(read-only): Только для чтения и чаще всего используют для отчетности или master-slave репликации.

1. **Способы создания представлений**

Через оператора(частый способ): CREATE VIEW  
Пример:

CREATE VIEW VW\_FilmsByGenre AS

SELECT f.FilmName, g.GenreName

FROM Films f

JOIN Genres g ON f.GenreID = g.GenreID;

Через обновление данных: INSERT, UPDATE, DELETE  
Пример:

CREATE VIEW VW\_ClientsInfo AS

SELECT ClientID, FullName, Phone

FROM Clients

WITH CHECK OPTION;

Временный запрос:

SELECT \*

FROM (

SELECT FilmName, ReleaseDate

FROM Films

WHERE YEAR(ReleaseDate) = 2025

) AS Films2025;