БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Отчет

«Лабораторная работа №5»

Лускиной Юлии Владимировны

студента 2 курса, 5 группы

специальность «Прикладная

математика»

Преподаватель:

Терех В.С.

Минск, 2025

**Разработка запросов**

1. Поэтапно по практикуму прошла все шаги.

Познакомилась с инструкциями запросов выборки, с агрегатными функциями, с вложенными запросами, с запросами действий и операторами выделения данных.

1. Выполнила самостоятельное задание.

За основу взяла разработанную базу данных из лабораторной работы № 4.

* 1. Написать запросы в СУБД MS SQL Server

− многотабличный запрос выборки с сортировкой и отбором данных (INNER JOIN, WHERE, ORDER BY),

SELECT

r.review\_id,

u.username,

m.title,

r.rating,

r.comment,

r.review\_date,

d.name AS director\_name,

d.last\_name AS director\_last\_name

FROM

Review r

INNER JOIN

Users u ON r.user\_id = u.user\_id

INNER JOIN

Movie m ON r.movie\_id = m.movie\_id

INNER JOIN

Directors d ON m.director\_id = d.director\_id

WHERE

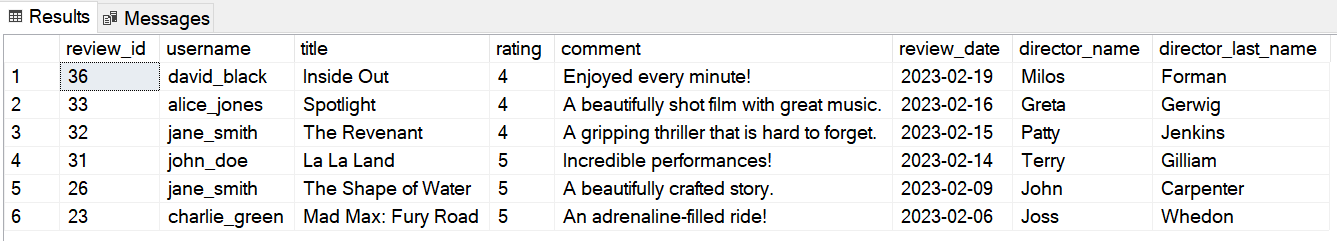
r.rating >= 3 -- Условие фильтрации по рейтингу

AND

m.release\_year >= 2015 -- Условие фильтрации по году выпуска фильма

ORDER BY

r.review\_date DESC; -- Сортировка по дате отзыва в порядке убывания



− запрос с применением вычисляемых полей (примечание: не применять агрегатные функции), например, вычислить возраст, полную цену, и типа подобное,

Чтобы осуществить это понадобилось добавить атрибут ‘birth\_date’, так же данные для него.

ALTER TABLE Users ADD birth\_date DATE;

UPDATE Users SET birth\_date = '1990-01-15' WHERE user\_id = 1;

UPDATE Users SET birth\_date = '1991-02-20' WHERE user\_id = 2;

UPDATE Users SET birth\_date = '1989-03-10' WHERE user\_id = 3;

UPDATE Users SET birth\_date = '1992-04-05' WHERE user\_id = 4;

UPDATE Users SET birth\_date = '1993-05-25' WHERE user\_id = 5;

. . .

UPDATE Users SET birth\_date = '1994-11-30' WHERE user\_id = 45;

UPDATE Users SET birth\_date = '1990-01-15' WHERE user\_id = 46;

UPDATE Users SET birth\_date = '2000-01-15' WHERE user\_id = 47;

UPDATE Users SET birth\_date = '2005-11-14' WHERE user\_id = 48;

UPDATE Users SET birth\_date = '2006-02-19' WHERE user\_id = 49;

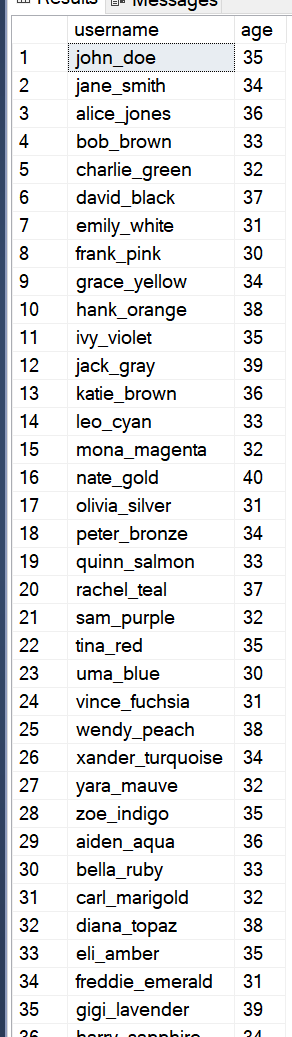
UPDATE Users SET birth\_date = '1990-07-15' WHERE user\_id = 50;

Запрос:

SELECT username,

YEAR(GETDATE()) - YEAR(birth\_date) AS age -- Вычисление возраста пользователя

FROM Users



− запрос выборки с внешним объединением двух отношений (LEFT|RIGHT JOIN),

SELECT

m.title,

m.genre,

r.rating

FROM

Movie AS m

LEFT JOIN

Review AS r ON m.movie\_id = r.movie\_id

WHERE

m.release\_year < 2013;



В конце есть значения NULL, так как в таблице Review нет отзывов на данные фильмы.

SELECT

m.title,

m.genre,

r.rating

FROM

Movie AS m

RIGHT JOIN

Review AS r ON m.movie\_id = r.movie\_id

WHERE

m.release\_year < 2013;

А если писать RIGHT JOIN, то эти значения с NULL не будут включаться в выборку.

− запрос группировкой, вычислением итогов и отбором данных (GROUP BY, HAVING),

SELECT

m.title,

AVG(r.rating) AS AverageRating

FROM

Movie AS m

JOIN

Review AS r ON m.movie\_id = r.movie\_id

GROUP BY

m.title

HAVING

AVG(r.rating) > 4.0

ORDER BY

AverageRating DESC;



− запрос на добавление (INSERT INTO),

INSERT INTO Movie (title, release\_year, genre, director\_id)

VALUES ('The New Adventure', 2023, 'Action', 1);

− запрос на удаление,

DELETE FROM Review

WHERE review\_id = 38;

− запрос на обновление,

Производила данный запрос в пункте выше.

UPDATE Users SET birth\_date = '1990-01-15' WHERE user\_id = 1;

− запрос на создание новой таблицы на основе существующей,

SELECT \*

INTO ArchivedMovies

FROM Movie

WHERE release\_year < 2000; -- Условие для выборки старых фильмов



− запрос на объединение (UNION), где применимо (только в SQL),

SELECT title, release\_year, genre

FROM Movie

UNION

SELECT title, release\_year, genre

FROM ArchivedMovies;

Команда UNION в SQL используется для объединения результатов двух или более запросов SELECT. Важно, чтобы каждый из объединяемых запросов имел одинаковое количество столбцов с совместимыми типами данных.

− вложенный запрос на SQL (вложение во фразе WHERE);

SELECT title

FROM Movie

WHERE movie\_id IN (

SELECT movie\_id

FROM Review

GROUP BY movie\_id

HAVING AVG(rating) > 3.0

);



− запрос на создание новой таблицы,

CREATE TABLE Movie2 (

movie\_id INT PRIMARY KEY, -- Уникальный идентификатор фильма

title VARCHAR(255) NOT NULL, -- Название фильма

release\_year INT, -- Год выпуска

genre VARCHAR(100), -- Жанр

director VARCHAR(100), -- Режиссёр

created\_at DATETIME DEFAULT GETDATE() -- Дата и время создания записи

);

− запрос на создание индекса,

CREATE INDEX idx\_title ON Movie (title);

− запрос на создание представления, объединяющего данные двух таблиц;

Представление в SQL (или View) — это виртуальная таблица, основанная на результатах запроса на выборку из одной или нескольких таблиц.

Представления не хранят данные сами по себе, а лишь содержат SQL-запрос, который выполняется каждый раз, когда вы обращаетесь к этому представлению.

CREATE VIEW MovieReviews AS

SELECT

m.movie\_id,

m.title,

m.release\_year,

m.genre,

r.rating,

r.comment

FROM

Movie m

JOIN

Review r ON m.movie\_id = r.movie\_id;



Хранится тут

