

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №5  
з дисципліни «Бази даних»**

**на тему: «Підзапити. Табличні вирази»**

Виконав:

Студент гр. ПЗ2011

Кулик С.В.

Прийняв:

Іванов О.П.

Дніпро, 2023

**Тема.** Підзапити. Табличні вирази

**Мета.** Отримати навички написання запитів із використанням підзапитів мови SQL. Познайомитися з табличними виразами, їх різновидами, вивчити сферу застосування табличних виразів.

**Скріншот діаграми бази даних із середовища SQL Server Management Studio**



**Файл Subqueries.sql**

use CinemaDB

go

-- Запит для отримання фільмів, які мають найновішу дату виходу

SELECT title

FROM dbo.Film

WHERE release\_date = (

SELECT MAX(release\_date)

FROM dbo.Film

);

-- Запит для отримання фільмів за певним жанром

SELECT title

FROM dbo.Film

WHERE genre IN (

SELECT name

FROM dbo.Genre

WHERE name LIKE '%Action%'

);

-- Запит для отримання користувачів, які купили квитки

SELECT username

FROM dbo.Users u

WHERE EXISTS (

SELECT 1

FROM s.Ticket t

WHERE t.users\_id = u.id

);

-- Запит для отримання назв фільмів, які мають високий рейтинг у відгуках (рейтинг > 8)

SELECT title

FROM dbo.Film f

WHERE EXISTS (

SELECT 1

FROM s.Review r

WHERE r.film\_id = f.id

AND r.rating > 8

);

-- Запит для отримання фільмів, тривалість яких перевищує середню тривалість фільмів у їхньому жанрі

SELECT title

FROM dbo.Film f

WHERE duration > (

SELECT AVG(duration)

FROM dbo.Film

WHERE f.genre = genre

);

-- Запит для отримання фільмів, тривалість яких перевищує середню тривалість у їхньому жанрі

SELECT title

FROM dbo.Film f

WHERE duration > ANY (

SELECT AVG(duration)

FROM dbo.Film

WHERE f.genre = genre

);

-- Вибірка фільмів, для яких існують відгуки користувачів

SELECT title

FROM dbo.Film f

WHERE EXISTS (

SELECT 1

FROM s.Review r

WHERE r.film\_id = f.id

);

-- Вибірка фільмів та кількості відгуків для кожного фільму

SELECT f.title, ReviewsPerFilm.TotalReviews

FROM dbo.Film f

JOIN (

SELECT film\_id, COUNT(\*) AS TotalReviews

FROM s.Review

GROUP BY film\_id

) AS ReviewsPerFilm ON f.id = ReviewsPerFilm.film\_id;

-- Вибірка фільмів та середньої оцінки за них

SELECT f.title, AVG(r.rating) AS AverageRating, HighestRatings.HighestRating

FROM dbo.Film f

JOIN s.Review r ON f.id = r.film\_id

JOIN (

SELECT film\_id, MAX(rating) AS HighestRating

FROM s.Review

GROUP BY film\_id

) AS HighestRatings ON f.id = HighestRatings.film\_id

GROUP BY f.title, HighestRatings.HighestRating;

-- Вибірка фільмів та середньої оцінки за них за допомогою двох CTE

WITH FilmRatings AS (

SELECT film\_id, AVG(rating) AS AverageRating

FROM s.Review

GROUP BY film\_id

),

TopRatedFilms AS (

SELECT f.title, fr.AverageRating

FROM dbo.Film f

JOIN FilmRatings fr ON f.id = fr.film\_id

WHERE fr.AverageRating > 8

)

SELECT \* FROM TopRatedFilms;

-- Вибірка середньої ціни квитків та кількості користувачів для кожного сеансу з використанням двох CTE

WITH AverageTicketPrice AS (

SELECT session\_id, AVG(price) AS AvgPrice

FROM s.Ticket

GROUP BY session\_id

),

UserCountPerSession AS (

SELECT session\_id, COUNT(DISTINCT users\_id) AS UserCount

FROM s.Ticket

GROUP BY session\_id

)

SELECT atp.session\_id, atp.AvgPrice, ucp.UserCount

FROM AverageTicketPrice atp

JOIN UserCountPerSession ucp ON atp.session\_id = ucp.session\_id;

-- Запит для створення зашифрованого View, який містить інформацію про фільми.

CREATE VIEW EncryptedView

WITH ENCRYPTION

AS

SELECT id, title, genre

FROM dbo.Film;

-- Повертає всі назви фільмів разом з часом початку сеансу для кожного фільму.

SELECT Film.title, Sesions.start\_time

FROM dbo.Film

CROSS APPLY (

SELECT \*

FROM dbo.Sesions

WHERE Film.id = dbo.Sesions.film\_id

) AS Sesions;

-- Повертає всі назви фільмів разом з часом початку сеансу для кожного фільму, включаючи фільми, для яких немає сеансів.

SELECT Film.title, Sesions.start\_time

FROM dbo.Film

OUTER APPLY (

SELECT \*

FROM dbo.Sesions

WHERE Film.id = Sesions.film\_id

) AS Sesions;

**Висновок**

Підзапити та табличні вирази є потужними інструментами у мові SQL, які дозволяють виконувати більш складні та контекстні запити до бази даних.

Підзапити (Subqueries) використовуються для отримання даних із інших запитів або підтаблиць, що використовуються в головному запиті. Зазвичай вони використовуються у виразах WHERE, HAVING або SELECT, і можуть бути корельованими (залежать від даних головного запиту) та самостійними (не залежать від головного запиту).

Табличні вирази (Table Expressions) включають такі концепції, як Common Table Expressions (CTEs), похідні таблиці, тимчасові таблиці та вирази VALUES. Вони дозволяють створювати тимчасові таблиці або підзапити для подальшого використання в запитах. CTEs надають зручну можливість іменування та організації підзапитів для полегшення розуміння запитів.

Ці інструменти застосовуються для виконання різних завдань, включаючи фільтрацію даних, об'єднання таблиць, підрахунки, сортування та інші операції. Вони корисні при створенні звітів, вибірці даних, побудові ієрархій, агрегації та інших аналітичних завдань.

Знання та розуміння підзапитів та табличних виразів дозволяє розробникам більш ефективно використовувати SQL для вирішення різних завдань у роботі з базами даних. Ці інструменти допомагають покращити читабельність запитів та зменшити повторюваний код, що робить роботу з даними більш продуктивною.