МIНIСТЕРСТВО ВИЩОЇ ОСВIТИ УКРАЇНИ НАЦIОНАЛЬНИЙ ТЕХНІЧНИЙ УНIВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Проєктування та розробка баз дани[х](https://classroom.google.com/u/0/c/NTg4NzUwODE5ODI0)

**“**Вивчення розширених можливостей мови SQL**”**

Виконала: Аксьонова Яна

Група: ТА-13, ННІАТЕ

Перевірив: Любицький С.В.

Київ 2023

**Завдання:** показати розширені можливості мови SQL, використовуючи базу даних, створену у попередніх роботах.

1. **Приклад створення і використання View.**

CREATE VIEW cinema\_with\_more\_then\_100\_workerquantity AS

SELECT \* FROM cinema

WHERE CinemaWorkerQuantity > 100

SELECT \* FROM cinema\_with\_more\_then\_100\_workerquantity

# Приклад збереженої процедури з параметрами за замовчуванням та умовними операторами.

CREATE PROCEDURE how\_many\_cinemaWorker @id INT = -1

AS BEGIN

IF @id = -1

SELECT 'Введіть кіно про яке ви хочете дізнатись'

END

ELSE

SELECT COUNT(\*) FROM Cinema

JOIN CinemaWorker

ON CinemaWorker.CinemaWorkerId = Cinema.CinemaId

EXEC how\_many\_cinemaWorker;

DECLARE @cinema\_id INT = 2

EXEC how\_many\_cinemaWorker @cinema\_id;

# Приклад збереженої процедури з параметрами що повертаються.

CREATE PROCEDURE film\_sponsore @film\_id INT

AS BEGIN

END

SELECT Film.FilmName, Film.Duration, Studio.StudioCountry FROM Film JOIN Sponsore

ON Sponsore.Film = Film.FilmId JOIN Studio

ON Studio.StudioId = Sponsore.Studio WHERE Film.FilmId = @film\_id

# EXEC film\_sponsore 2;

# Приклад збереженої процедури в якій використовується робота з курсором.

CREATE PROCEDURE get\_all\_cinemaWorker\_position AS

BEGIN

DECLARE @position VARCHAR(50);

DECLARE s\_cursor CURSOR FOR SELECT position FROM CinemaWorker; OPEN s\_cursor;

FETCH NEXT FROM s\_cursor INTO @position; WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT @position;

FETCH NEXT FROM s\_cursor INTO @position; END;

CLOSE s\_cursor; DEALLOCATE s\_cursor;

END;

# EXEC get\_all\_cinemaWorker\_position;

# Приклад функції з параметрами що повертає скалярне значення.

CREATE FUNCTION get\_seanse\_count (@HallId INT)

RETURNS FLOAT AS

BEGIN

DECLARE @count INT;

SELECT @count = COUNT(\*) FROM Hall

JOIN Seanse

ON Seanse.Hall = Hall.HallId WHERE @HallId = Hall.HallId;

RETURN @count;

END;

# SELECT dbo.get\_seanse\_count(1);

# Приклад функції що повертає табличне значення.

CREATE FUNCTION get\_sponsors (@FilmId INT)

RETURNS TABLE AS

RETURN (

SELECT Film.\*, Studio.StudioCountry FROM Film JOIN Sponsore

ON Sponsore.Film = Film.FilmId JOIN Studio

ON Studio.StudioId = Sponsore.Studio WHERE Film.FilmId = @FilmId

);

SELECT \* FROM dbo.get\_sponsors(1);

# Приклад тригера що використовує змінні inserted та deleted.

CREATE TABLE insertedStudioLog( StudioId INT,

StudioCountry INT, insertedDate DATETIME)

CREATE TRIGGER tr\_studio\_inserted ON Studio

AFTER INSERT

AS BEGIN

INSERT INTO insertedStudioLog (StudioId, StudioCountry, insertedDate) SELECT i.StudioId, i.StudioCountry, GETDATE()

FROM inserted i; END;

CREATE TABLE deletedStudioLog( StudioId INT,

StudioCountry INT, deletedDate DATETIME)

CREATE TRIGGER tr\_studio\_deleted ON Studio

AFTER DELETE AS

BEGIN

INSERT INTO deletedStudioLog (StudioId, StudioCountry, deletedDate) SELECT d.StudioId, d.StudioCountry, GETDATE()

FROM deleted d; END;

# Приклад тригера INSTEAD OF.

ALTER TABLE Cinema ADD is\_deleted INT

CREATE TRIGGER CinemaDelete ON Cinema

INSTEAD OF DELETE AS

UPDATE Cinema SET is\_deleted = 1

WHERE CinemaId =(SELECT CinemaId FROM deleted)

Завдання

CREATE FUNCTION GetAvailableSeatsCount(@hallID int) RETURNS int

AS

BEGIN

DECLARE @totalSeats int;

DECLARE @bookedSeats int;

DECLARE @availableSeats int;

SELECT @totalSeats = SeatsQuantity FROM Hall WHERE HallID = @hallID;

SELECT @bookedSeats = COUNT(\*) FROM Ticket

INNER JOIN Seanse ON Ticket.Seanse = Seanse.SeanseID

WHERE Seanse.Hall = @hallID;

SET @availableSeats = @totalSeats - @bookedSeats;

RETURN @availableSeats;

END;

DECLARE @hallID int = 1;

SELECT dbo.GetAvailableSeatsCount(@hallID) AS AvailableSeatsCount;

Зображення, що містить текст, Шрифт, ряд, знімок екрана

Автоматично згенерований опис

**Висновок**: у результаті виконаної роботи я використав розширенні можливості мови SQL. Створив процедури та функції, навчився використовувати View, а також освоїм роботу з тригерами та курсором у SQL.