Міністерство освіти і науки України Харківський національний університет радіоелектроніки Кафедра програмної інженерії

Лабораторна робота № 1 з дисципліни «Серверні рішення баз даних» з теми: «ЗБЕРЕЖЕНІ ПРОЦЕДУРИ ТА ФУНКЦІЇ Т-SQL»

Перевірила старший викладач кафедри ПІ Широкоптелєва М.С Виконали ст. групи ПЗПІ-22-7 Голоха Н. Е. Колесник О.А.

1 ЗБЕРЕЖЕНІ ПРОЦЕДУРИ ТА ФУНКЦІЇ T-SQL

Мета роботи:

Ознайомлення з основними можливостями мови T-SQL. Закріплення теоретичних і набуття практичних навичок роботи з вбудованими функціями мови T-SQL, а також створення збережених, призначених для користувача функцій і обробка результатів їх виконання.

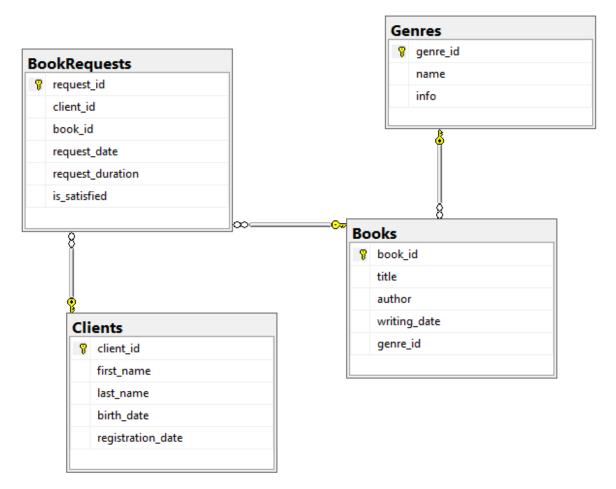


Рисунок 1 - Схема створенної БД

4) Розробіть функцію, яка за назвою товару (вхідний параметр) повертає дані щодо найбільшого замовлення цього товару (якщо існує декілька таких замовлень, тоді виводити інформацію про всі такі замовлення). За відсутністю товару або замовлень цього товару вивести «Відсутні незадоволені запити для книги«назва книги»;

Найдовше невиконане замовлення.

```
GO
CREATE FUNCTION GetLongest(@bookTitle NVARCHAR(255))
RETURNS @ResultsTable TABLE (
    ClientName NVARCHAR(255),
```

```
BookTitle NVARCHAR (255),
   RequestDate DATE,
   RequestDuration INT
AS
BEGIN
   DECLARE @maxDuration INT;
   SELECT @maxDuration = MAX(r.request duration)
   FROM BookRequests r
    JOIN Books b ON r.book id = b.book id
   WHERE b.title = @bookTitle AND r.is satisfied = 0;
   IF @maxDuration IS NULL
   BEGIN
        INSERT INTO @ResultsTable (ClientName, BookTitle, RequestDate,
RequestDuration)
       VALUES ('Відсутні незадоволені запити для книги "' + @bookTitle + '"',
NULL, NULL, NULL);
       RETURN;
   END;
    INSERT INTO @ResultsTable (ClientName, BookTitle, RequestDate, RequestDuration)
    SELECT CONCAT(c.first name, ' ', c.last name) AS ClientName,
          b.title AS BookTitle,
          r.request date AS RequestDate,
           r.request duration AS RequestDuration
    FROM BookRequests r
    JOIN Books b ON r.book_id = b.book_id
    JOIN Clients c ON r.client id = c.client id
   WHERE b.title = @bookTitle AND r.is_satisfied = 0 AND r.request_duration =
@maxDuration;
   RETURN;
END;
GO
```

	ClientName	Book Title	Request Date	Request Duration
1	John Doe	The Hobbit	2023-09-15	14
2	Jane Smith	The Hobbit	2024-09-17	14

5) Розробіть функцію, яка за назвою товару (вхідний параметр) повертає дані щодо найбільшого замовлення цього товару (якщо існує декілька таких замовлень, тоді — одного, будь-якого з них). За відсутністю товару або замовлень цього товару вивести «Відсутні незадоволені запити для книги«назва книги»;

```
Go
CREATE FUNCTION GetLongest(@bookTitle NVARCHAR(255))
RETURNS NVARCHAR(MAX)
AS
BEGIN
DECLARE @result NVARCHAR(MAX);

-- Знаходимо найдовший незадоволений запит для конкретної книги
SELECT TOP 1 @result = CONCAT(
```

```
'Клієнт: ', c.first_name, ' ', c.last_name,
        ', Назва книги: ', b.title,
        ', Дата запиту: ', r.request_date,
        ', Тривалість: ', r.request_duration, ' днів')
    FROM BookRequests r
    JOIN Books b ON r.book_id = b.book_id
    JOIN Clients c ON r.client_id = c.client_id
    WHERE b.title = @bookTitle AND r.is_satisfied = 0
    IF @result IS NULL
    BEGIN
       RETURN CONCAT(N'Biдсутні незадоволені запити для книги «', @bookTitle,
N'»');
    END;
   RETURN @result;
END;
GO
SELECT dbo.GetLongest('The Hobbit');
Drop Function GetLongest;
```

```
(No column name)

1 Клієнт: John Doe, Назва книги: The Hobbit, Дата запиту: 2023-09-15, Тривалість: 14 днів
```

13) Розробіть процедуру, що для заданого відділу в полів Іпбо записує поточну дату назву виробника, у якого найбільше різних товарів у цьому відділі (якщо таких виробників декілька, тоді будь-якого з них). Запис в жанр автора у якого більше всього книг в жанрі.

```
CREATE PROCEDURE UpdateGenreAuthorInfo
BEGIN
    DECLARE @genre id INT;
    DECLARE @max_author NVARCHAR(255);
    DECLARE genre cursor CURSOR FOR
    SELECT genre_id FROM Genres;
    OPEN genre cursor;
    FETCH NEXT FROM genre cursor INTO @genre id;
   WHILE @@FETCH STATUS = 0
    BEGIN
        SELECT TOP 1 @max_author = author
        FROM Books
        WHERE genre id = @genre id
        GROUP BY author
        ORDER BY COUNT (book id) DESC;
        IF @max_author IS NOT NULL
        BEGIN
           UPDATE Genres
            SET info = @max author
            WHERE genre id = @genre id;
```

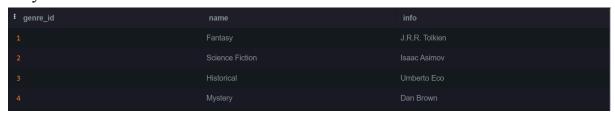
```
END
ELSE
BEGIN

UPDATE Genres
SET info = NULL
WHERE genre_id = @genre_id;
END

FETCH NEXT FROM genre_cursor INTO @genre_id;
END;

CLOSE genre_cursor;
DEALLOCATE genre_cursor;
END;

EXEC UpdateGenreAuthorInfo
DROP PROCEDURE IF EXISTS UpdateGenreAuthorInfo;
```



14) Розробіть процедуру, що видаляє співробітників (крім начальника відділу – в полі Information вказано «Мападег»), що працюють у відділах, в яких сумарний обсяг продажів менше заданого видаляє клієнтів у яких не було запитів більше п(вводиться) днів.

```
CREATE PROCEDURE DeleteInactiveClients
    @days INT -- Вводится количество дней

AS

BEGIN
    DELETE FROM Clients
    WHERE client_id NOT IN (
        SELECT DISTINCT client_id
        FROM BookRequests
        WHERE DATEDIFF(DAY, request_date, GETDATE()) <= @days
    );

END;

EXEC DeleteInactiveClients @days = 1

DROP PROCEDURE IF EXISTS DeleteInactiveClients;
```



Створення бд:

```
-- Таблиця з жанрами
CREATE TABLE Genres (
   genre id INT PRIMARY KEY IDENTITY(1,1),
   name NVARCHAR(255) UNIQUE NOT NULL,
   info NVARCHAR(255) -- Поле для запису інформації про автора з найбільшою
кількістю книг
);
-- Таблиця з книгами
CREATE TABLE Books (
   book id INT PRIMARY KEY IDENTITY (1,1),
   title NVARCHAR (255) NOT NULL,
   author NVARCHAR (255) NOT NULL,
   writing date DATE NOT NULL,
   genre id INT,
    FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)
);
-- Таблиця з клієнтами
CREATE TABLE Clients (
   client_id INT PRIMARY KEY IDENTITY(1,1),
   first_name NVARCHAR(255) NOT NULL,
   last_name NVARCHAR(255) NOT NULL,
   birth date DATE NOT NULL,
    registration date DATE NOT NULL
);
-- Таблиця запитів на отримання книг
CREATE TABLE BookRequests (
    request id INT PRIMARY KEY IDENTITY(1,1),
   client_id INT,
   book_id INT,
    request_date DATE NOT NULL,
    request_duration INT,
    is satisfied BIT,
    FOREIGN KEY (client id) REFERENCES Clients(client id) ON DELETE SET NULL,
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
```

```
INSERT INTO Genres (name, info)
VALUES
('Fantasy', NULL),
('Science Fiction', NULL),
('Historical', NULL),
('Mystery', NULL);
INSERT INTO Books (title, author, writing date, genre id)
VALUES
('The Hobbit', 'J.R.R. Tolkien', '1937-09-21', 1),
('The Fellowship of the Ring', 'J.R.R. Tolkien', '1954-07-29', 1),
('Dune', 'Frank Herbert', '1965-08-01', 2),
('Foundation', 'Isaac Asimov', '1951-05-01', 2),
('The Name of the Rose', 'Umberto Eco', '1980-11-01', 3),
('The Da Vinci Code', 'Dan Brown', '2003-03-18', 4);
INSERT INTO Clients (first_name, last_name, birth_date, registration_date)
VALUES
('John', 'Doe', '1990-01-15', '2020-08-05'),
('Jane', 'Smith', '1985-06-20', '2021-03-22'),
('Alice', 'Johnson', '1995-12-10', '2019-11-11'),
('Bob', 'Williams', '2000-04-25', '2022-01-30');
INSERT INTO BookRequests (client id, book id, request date, request duration,
is satisfied)
VALUES
(1, 1, '2023-09-15', 14, 0),
(2, 1, '2024-09-17', 14, 0),
(2, 2, '2023-09-18', 30, 0),
(3, 3, '2023-09-20', 7, 1),
(4, 4, '2023-09-25', 14, 0),
(1, 5, '2023-09-28', 21, 1);
select * from Genres
select * from Books
select * from Clients
select * from BookRequests
Drop table Genres
Drop table Books
Drop table Clients
Drop table BookRequests
```

Висновки

Отже у ході роботи ми ознайомилися з основними можливостями мови T-SQL. Закріпили теоретичні і набули практичних навичок роботи з вбудованими функціями мови T-SQL, а також створення збережених, призначених для користувача функцій і обробки результатів їх виконання.