**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Изображение выглядит как текст, старый, винтажный, нарисованный

Автоматически созданное описание**Институт среднего профессионального образования**

**ОТЧЕТ**

**По выполнению лабораторных работ**

**по предмету «МДК 11.01 Технология разработки и защиты БД»**

**Выполнил:** Лелюков Матвей Алексеевич

**Группа №** 32919/22

**Преподаватель:** Сынкова Алина Денисовна

**« »** 20 Оценка:

г. Санкт-Петербург

2025

**Цель работы:** Изучение назначения и типов курсоров, синтаксиса языка T – SQL для создания и открытия курсоров, выборки данных из курсора и изменения строк таблиц с помощью курсоров, удаления данных, закрытия и освобождения курсоров, а также приобретения навыков их применения и управления с помощью команд и системных хранимых процедур SQL Server.

1. **Создайте таблицу Clients со следующими атрибутами: Id, Name, Address.**

Создадим и заполним таблицу Clients:

CREATE TABLE Clients (

ID INT PRIMARY KEY,

Name NVARCHAR(255),

Address NVARCHAR(MAX),

);

INSERT INTO Clients (ID, Name, Adress) VALUES

(1, 'Светлана', 'Москва, Большая Лубянка 2'),

(2, 'Николай', 'Санкт-Петербург, Елизаветинская 9-15'),

(3, 'Алена', 'Тамбов, Лесной пер 7'),

(4, 'Михаил', 'Воронеж, Шоссейная 13'),

(5, 'Алексей', 'Псков, Дачная 8')

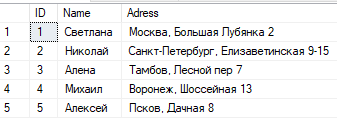


Рис 1 – таблица «Clients»

1. **С помощью курсора создайте запрос, который будет выводить информацию о всех клиентах c нечетным Id.**

Код курсора:

DECLARE @Id INT, @Name VARCHAR(MAX), @Address VARCHAR(MAX)

DECLARE mega\_cursor CURSOR FAST\_FORWARD

FOR

SELECT \*

FROM Clients

WHERE Id % 2 = 1

OPEN mega\_cursor

FETCH NEXT FROM mega\_cursor INTO @Id, @Name, @Address

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT CONCAT('Client ID: ', @Id, ', Имя: ', @Name, ', Адрес: ', ISNULL(@Address, 'Нет адреса'))

FETCH NEXT FROM mega\_cursor INTO @Id, @Name, @Address

END

CLOSE mega\_cursor

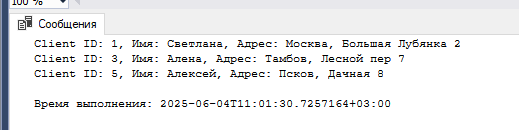


Рис 2 – курсор «mega\_cursor»

1. **Создайте таблицу OlympMath с результатами Олимпиады по математике за 2019, 2020 и 2021 года. Таблица OlympMath состоит из следующих атрибутов: id, Name, res19, res20, res21.**

Код запроса:

CREATE TABLE OlympMath (

id INT PRIMARY KEY,

Name VARCHAR(25) NOT NULL,

res19 INT,

res20 INT,

res21 INT

)

INSERT INTO OlympMath (id, Name, res19, res20, res21)

VALUES

(1, 'Светлана', 90, 95, 100),

(2, 'Антон', 70, 75, 80),

(3, 'Тимофей', 60, 65, 70),

(4, 'Николай', 50, 55, 60),

(5, 'Артем', 40, 45, 50)

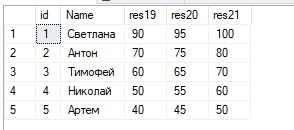


Рис 3 – таблица «OlympMath»

1. **С помощью курсора создайте запрос, который будет находить и выводить лучший результат для каждого участника. То есть максимальный в строке.**

Код курсора:

DECLARE @id INT, @Name VARCHAR(25)

DECLARE olymp\_cursor CURSOR FAST\_FORWARD

FOR

SELECT id, Name

FROM OlympMath

OPEN olymp\_cursor

FETCH NEXT FROM olymp\_cursor INTO @id, @Name

WHILE @@FETCH\_STATUS = 0

BEGIN

DECLARE @max\_result INT

SELECT @max\_result = MAX(result)

FROM (

SELECT res19 AS result FROM OlympMath WHERE id = @id

UNION

SELECT res20 FROM OlympMath WHERE id = @id

UNION

SELECT res21 FROM OlympMath WHERE id = @id

) AS results

PRINT CONCAT('ID: ', @id, ', Имя: ', @Name, ', Лучший результат: ', ISNULL(CAST(@max\_result AS VARCHAR), 'No results available'))

FETCH NEXT FROM olymp\_cursor INTO @id, @Name

END

CLOSE olymp\_cursor

DEALLOCATE olymp\_cursor

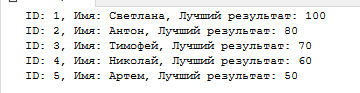


Рис 4 – курсор «olymp\_cursor»

1. **Создайте две таблицы со следующей структурой (папки и файлы):**

Код полного запроса:

CREATE TABLE Folders (

Id INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

DateCreate VARCHAR(50) NOT NULL,

Parent INT NULL,

FOREIGN KEY (Parent) REFERENCES Folders(Id)

);

CREATE TABLE Files (

Id INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

ext VARCHAR(10),

Date\_Create VARCHAR(50) NOT NULL,

FileSize INT,

FolderId INT NOT NULL,

FOREIGN KEY (FolderId) REFERENCES Folders(Id)

);

INSERT INTO Folders (Id, Name, DateCreate, Parent) VALUES

(1, 'Загрузки', '2025-03-21', NULL),

(2, 'Рабочий стол', '1952-06-01', 1),

(3, 'Музыка', '1254-12-12', 1),

(4, 'Документы', '1496-01-01', 1),

(5, 'Фотографии', '1555-01-01', 3);

INSERT INTO Files (Id, Name, ext, Date\_Create, FileSize, FolderId) VALUES

(1, 'книга', 'pdf', '1969-04-06', 115, 2),

(2, 'дабстеп', 'mp3', '2022-01-21', 5, 3),

(3, 'документ', 'txt', '2023-01-01', 50, 4),

(4, 'вирус', 'exe', '2025-06-20', 500, 1),

(5, 'светлана', 'png', '2000-01-01', 25, 5);

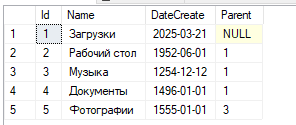


Рис 5 – таблица «Folders»

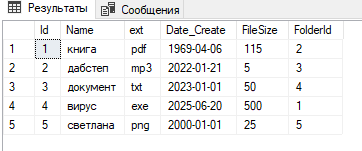


Рис 6 – таблица «files»

1. **Реализуйте функцию, которая будет принимать на входе идентификатор каталога и выводить все файлы и каталоги, которые находятся в заданном каталоге и во всех подкаталогах.**

Код функции:

CREATE FUNCTION GetAll (@FolderId INT)

RETURNS @Result TABLE (ItemType VARCHAR(10), Id INT, Name VARCHAR(255), Ext VARCHAR(10), Date\_Create VARCHAR(50), FileSize INT, FolderId INT)

AS

BEGIN

WITH FolderCTE AS (

SELECT Id, Name, DateCreate, Parent

FROM Folders

WHERE Id = @FolderId

UNION ALL

SELECT f.Id, f.Name, f.DateCreate, f.Parent

FROM Folders f

INNER JOIN FolderCTE fc ON f.Parent = fc.Id

)

INSERT INTO @Result

SELECT 'Folder', Id, Name, NULL, DateCreate, NULL, Parent

FROM FolderCTE

UNION ALL

SELECT 'File', f.Id, f.Name, f.ext, f.Date\_Create, f.FileSize, f.FolderId

FROM Files f

WHERE f.FolderId IN (SELECT Id FROM FolderCTE);

RETURN;

END;

Проверим корректность работы функции, выведя все файлы из папки «Рабочий стол» и «Фотографии»:

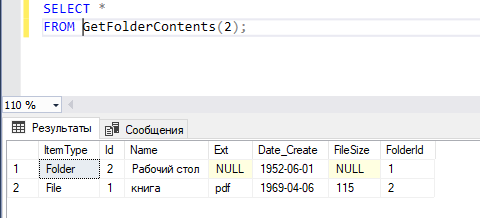


Рис 7 – вывод содержимого рабочего стола

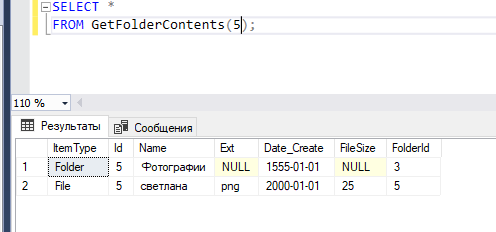


Рис 8 – вывод содержимого папки фотографий