-- Laboratoire 12 & 13: 10% de la note globale

-- 1. À l’aide d’une procédure stockée et en utilisant des variables

-- a. Créer une base de données (magasin)

-- b. Créer les 4 tables suivantes:

-- Client Produit client\_log produit\_log

-- Id\_client Id\_produit Id\_client Id\_produit

-- Nom Nom Date\_insertion Date\_insertion

-- Prénom Prix Description Description

-- Email

-- Création de la procédure stockée

CREATE PROCEDURE CreateMagasinDatabaseAndTables

AS

BEGIN

IF NOT EXISTS (SELECT \* FROM sys.databases WHERE name = 'magasin')

BEGIN

CREATE DATABASE magasin;

END

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Client' AND type = 'U')

BEGIN

CREATE TABLE Client (

Id\_client INT IDENTITY(1,1) PRIMARY KEY,

Nom VARCHAR(50) NOT NULL,

Prenom VARCHAR(50) NOT NULL,

Email VARCHAR(320) UNIQUE NOT NULL,

CONSTRAINT UC\_Client UNIQUE (Id\_client)

);

END

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'Produit' AND type = 'U')

BEGIN

CREATE TABLE Produit (

Id\_produit INT IDENTITY(1,1) PRIMARY KEY,

Nom VARCHAR(50) NOT NULL,

Prix DECIMAL(10,2) NOT NULL CHECK (Prix > 0),

CONSTRAINT UC\_Produit UNIQUE (Id\_produit)

);

END

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'client\_log' AND type = 'U')

BEGIN

CREATE TABLE client\_log (

Id\_client INT FOREIGN KEY REFERENCES Client(Id\_client) ON DELETE CASCADE,

Date\_insertion DATETIME NOT NULL,

Description VARCHAR(50) NOT NULL,

CONSTRAINT UC\_client\_log UNIQUE (Id\_client)

);

END

IF NOT EXISTS (SELECT \* FROM sys.tables WHERE name = 'produit\_log' AND type = 'U')

BEGIN

CREATE TABLE produit\_log (

Id\_produit INT,

Date\_insertion DATETIME NOT NULL,

Description VARCHAR(50) NOT NULL,

CONSTRAINT UC\_produit\_log UNIQUE (Id\_produit)

);

END

END;

GO

Graphical user interface, application

Description automatically generated

-- c. Exécuter la procédure stockée

EXEC CreateMagasinDatabaseAndTables;

GO

Graphical user interface, text, application

Description automatically generated

-- 2. Créer un trigger sur la table client après insertion en insérant le id,

-- la date d’insertion et le mot ‘inséré’ dans les colonnes de la table client\_log.

-- Sélection de la base de données

USE magasin;

GO

-- Création du trigger sur la table Client

IF OBJECT\_ID('trg\_Client\_AfterInsert', 'TR') IS NOT NULL

DROP TRIGGER trg\_Client\_AfterInsert;

GO

CREATE TRIGGER trg\_Client\_AfterInsert

ON Client

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO client\_log (Id\_client, Date\_insertion, Description)

SELECT i.Id\_client, GETDATE(), 'inséré'

FROM inserted i;

END;

GO

-- Insérer un nouveau client

INSERT INTO Client (Nom, Prenom, Email)

VALUES ('Benslimane', 'Abderraouf', '2706014@collegelacite.ca')

GO

-- Vérifier l'insertion dans la table Client et client\_log

SELECT c.Id\_client, c.Nom, c.Prenom, c.Email, cl.Date\_insertion, cl.Description

FROM Client c

JOIN client\_log cl ON c.Id\_client = cl.Id\_client;

GO

Timeline

Description automatically generated

-- 3. Créer un trigger sur la table produit après suppression en insérant le id\_produit,

-- la date de suppression et le mot 'supprimé’ dans les colonnes de la table produit\_log.

-- Sélection de la base de données

USE magasin;

GO

-- Vérifier si le trigger existe, et le supprimer s'il existe

IF OBJECT\_ID('trg\_AfterDelete\_Produit', 'TR') IS NOT NULL

DROP TRIGGER trg\_AfterDelete\_Produit;

GO

-- Création du trigger après suppression sur la table Produit

CREATE TRIGGER trg\_AfterDelete\_Produit

ON Produit

AFTER DELETE

AS

BEGIN

DECLARE @id\_produit INT;

DECLARE @date\_suppression DATETIME;

DECLARE @description VARCHAR(50);

SELECT @id\_produit = Id\_produit FROM deleted;

SET @date\_suppression = GETDATE();

SET @description = 'supprimé';

INSERT INTO produit\_log (Id\_produit, Date\_insertion, Description)

VALUES (@id\_produit, @date\_suppression, @description);

END;

GO

-- Test du trigger

-- Insérer un produit pour le tester

INSERT INTO Produit (Nom, Prix)

VALUES ('Écouteurs Bluetooth', 49.99);

GO

-- Supprimer le produit pour déclencher le trigger

DELETE FROM Produit

WHERE Id\_produit = 1;

GO

-- Vérifier que le trigger a bien inséré une entrée dans produit\_log

SELECT \* FROM produit\_log;

GO

Timeline

Description automatically generated

-- 4. Insérer deux clients dans la table client et deux produits dans la table produit

-- Sélection de la base de données

USE magasin;

GO

-- Insérer deux clients dans la table Client

INSERT INTO Client (Nom, Prenom, Email)

VALUES ('Ait Ali', 'Laila', '2691473@collegelacite.ca'),

('Bessier', 'Loovens', '2703243@collegelacite.ca');

GO

-- Insérer deux produits dans la table Produit

INSERT INTO Produit (Nom, Prix)

VALUES ('Ordinateur portable 15"', 749.99),

('Carte graphique GTX 1050 Ti', 249.99);

GO

-- Vérifier l'insertion dans la table Client et client\_log

SELECT c.Id\_client, c.Nom, c.Prenom, c.Email, cl.Date\_insertion, cl.Description

FROM Client c

JOIN client\_log cl ON c.Id\_client = cl.Id\_client;

GO

-- Vérifier l'insertion dans la table Produit

SELECT \* FROM Produit;

GO

-- Vérifier l'insertion dans la table produit\_log

SELECT \* FROM produit\_log;

GO

Timeline

Description automatically generated

-- 5. Supprimer un produit

-- Sélection de la base de données

USE magasin;

GO

-- Supprimer le produit avec Id\_produit 2

DELETE FROM Produit

WHERE Id\_produit = 2;

GO

-- Vérifier l'insertion dans la table Produit

SELECT \* FROM Produit;

GO

-- Vérifier l'insertion dans la table produit\_log

SELECT \* FROM produit\_log;

GO

Text

Description automatically generated with medium confidence

-- 6. Afficher les éléments des tables client\_log et produit\_log

-- Sélection de la base de données

USE magasin;

GO

-- Afficher les éléments de la table client\_log

SELECT \* FROM client\_log;

GO

-- Afficher les éléments de la table produit\_log

SELECT \* FROM produit\_log;

GO

Graphical user interface, text

Description automatically generated