

Лабораторна робота №1

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ БД ЗАСОБАМИ MS SQL SERVER.АВТОМАТИЗАЦІЯ ПРОЦЕДУР ОБРОБКИ ДАНИХ.

Мета роботи: дослідження засобів реалізації БД та SQL в середовище MSSQL Server.

Хід роботи

1. Створити базу даних за індивідуальним варіантом.

```
DESKTOP-DE8O4J0\S...arket - Diagram_0 1(2).sql - DESKTOP...DE8O4J0\prime (53))
create database Supermarket
use Supermarket

create table Virib(
    id_Viroy int identity(1,1) not null primary key,
    Kod_viroby int,
    Imenyvannya varchar(40),
    Price money
)

create table Consumer(
    id_Consumer int identity(1,1) not null primary key,
    PIB varchar (60),
    Tel char(12) check (tel like '([0-9][0-9][0-9])[0-9][0-9][0-9][0-9][0-9][0-9]'),
    Addr varchar(100),
    PoshtIndex int,
    Rozrah_rah int
)

create table Vid_Dostavki(
    id_vid_dost int identity(1,1) not null primary key,
    Nazva_Dost varchar (60)
)

create table Supermarket(
    id_Supermarket int identity(1,1) not null primary key,
    Nazva varchar (60),
    Adresa varchar (100)
)

create table Supermarket_Virib(
    id_sup_vir int identity(1,1) not null primary key,
    n_viroby int references Virib(id_Viroy),
    n_supermark int references Supermarket(id_Supermarket)
)

create table Dogovir(
    id_dogovir int identity(1,1) not null primary key,
    Nomer_Dogovory int,
    Data_Pidpisannya date,
    Kilk_virobiv int,
    n_spozhyvacha int references Consumer(id_Consumer),
    n_viroby int references Virib(id_Viroy)
)

create table Dani_Dostavki(
    id_dani_dost int identity(1,1) not null primary key,
```

Рис. 1 Створення бази даних

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.5.000 – Лр.1						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Куленко І.В.			Звіт з лабораторної роботи №1			Лім.	Арк.	Аркуші	
Перевір.		Праздніков О.В.								1	10
Реценз.								ФІКТ, гр. КБк-22-1			
Н. Контр.											
Зав.каф.		Єфіменко А.А.									

DESKTOP-DE8O4J0\S...arket - Diagram_0 1(2).sql - DESKTOP...DE8O4J0\prime (53))

```

create table Dani_Dostavki(
    id_dani_dost int identity(1,1) not null primary key,

    No_mashini int,
    No_podorozh_lista int,
    Price_Avto money,

    No_ZD_kontainera int,
    No_ZD_kvitanicii int,
    Price_ZD money,

    No_reisa int,
    No_avia_kvitka int,
    Price_Avia money,

    n_vid_dost int references Vid_Dostavki(id_vid_dost)
)

create table TTN(
    id_ttn int identity(1,1) not null primary key,
    Kilkist_Vir int,
    Data_Vidprav int,
    n_dogovir int references Dogovir(id_dogovir),
    n_dani_dost int references Dani_Dostavki(id_dani_dost)
)

```

Рис. 2 Створення бази даних

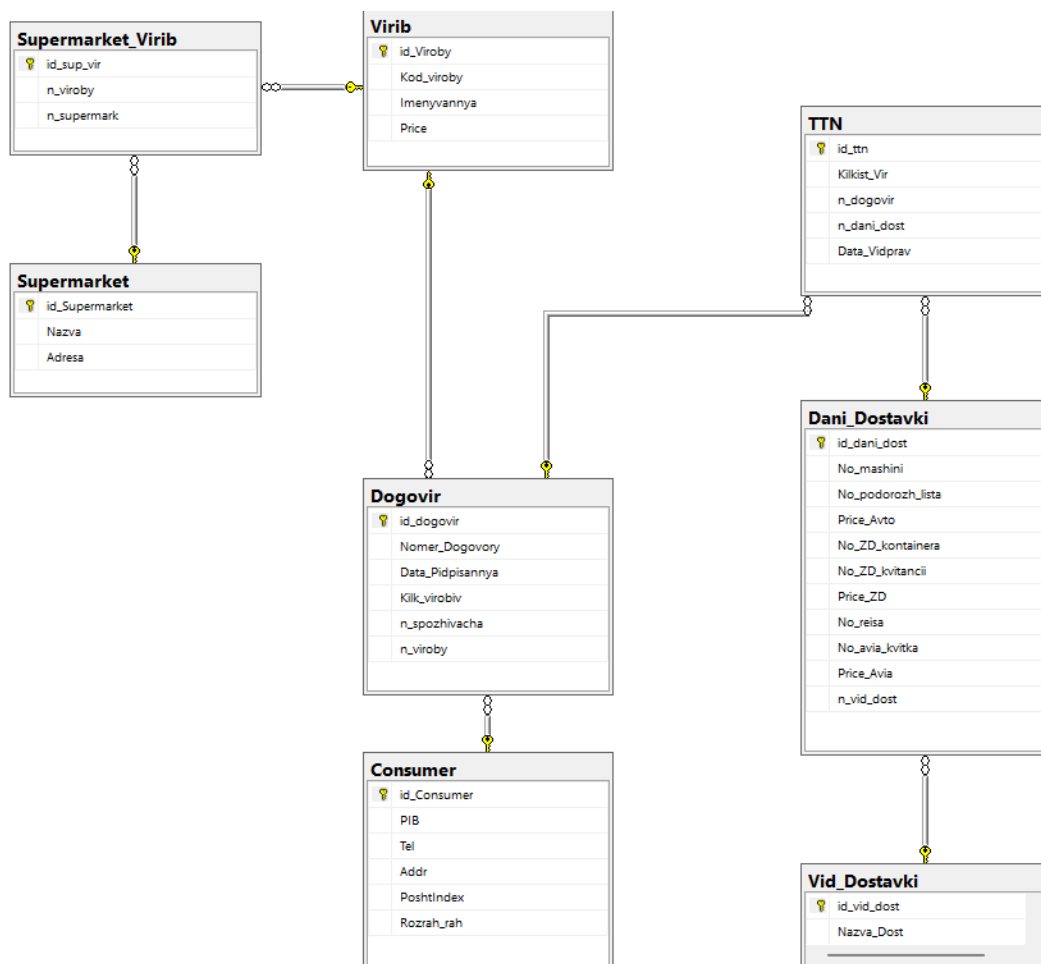


Рис. 3 Діаграма створеної бази даних

2. Внести текстові записи (3-4 записи)

```

-- Додати дані до таблиці Supermarket
INSERT INTO Supermarket (Nazva, Adresa)
VALUES
('Супермаркет 1', 'вул. Головна, 123'),
('Супермаркет 2', 'вул. Перша, 456'),
('Супермаркет 3', 'вул. Друга, 789');

-- Додати дані до таблиці Virib
INSERT INTO Virib (Kod_viroby, Imenyvannya, Price)
VALUES
(1, 'Хліб', 10.50),
(2, 'Семочки', 15.25),
(3, 'Сухарики', 20.00);

-- Додати дані до таблиці Supermarket_Virib
INSERT INTO Supermarket_Virib (n_viroby, n_supermark)
VALUES
(1, 1),
(2, 2),
(3, 3);

-- Додати дані до таблиці Consumer
INSERT INTO Consumer (PIB, Tel, Addr, PoshtIndex, Rozrah_rah)
VALUES
('Іванов Іван Іванович', '(063)4567890', 'вул. Центральна, 1', 12345, 1000),
('Петров Петро Петрович', '(098)6543210', 'вул. Периферійна, 5', 54321, 2000),
('Сидорова Марія Петрівна', '(097)2233344', 'вул. Сонячна, 10', 67890, 1500);

-- Додати дані до таблиці Dogovir
INSERT INTO Dogovir (Nomer_Dogovory, Data_Pidpisannya, Kilk_virobiv, n_spozhivacha, n_viroby)
VALUES
(1001, '2023-01-15', 5, 1, 1),
(1002, '2023-02-20', 3, 2, 2),
(1003, '2023-03-25', 4, 3, 3);

-- Додати дані до таблиці Vid_Dostavki
INSERT INTO Vid_Dostavki (Nazva_Dost)
VALUES
('Автотранспорт'),
('Залізничний транспорт'),
('Повітряний транспорт');

```

Рис. 4 Заповнення даними

```

-- Додати дані до таблиці Dani_Dostavki
INSERT INTO Dani_Dostavki (No_mashini, No_podorozh_list, Price_Avto, No_ZD_kontaynera, No_ZD_kvintancii, Price_ZD, No_reisa, No_avia_kvitka, Price_Avia, n_vid_dost)
VALUES
(101, 501, 500.00, null, null, null, null, null, null, 1),
(null, null, null, 201, 601, 900.00, null, null, null, 2),
(null, null, null, null, null, null, 301, 701, 2000.00, 3);

-- Додати дані до таблиці TTN
INSERT INTO TTN (Kilkist_Vir, Data_Vidprav, n_dogovir, n_dani_dost)
VALUES
(5, '2023-01-24', 1, 1),
(2, '2023-03-06', 2, 2),
(4, '2023-03-27', 3, 3);

```

Рис. 5 Заповнення даними

3. Створити представлення та процедури обробки даних у відповідності до індивідуального завдання.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.5.000 – Лр.1	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

--Індивідуальні завдання за варіантом 5
--Формування списку виробів і їхніх споживачів.
select Consumer.PIB, Virib.Imenyvannya, Virib.Price, Dogovir.Kilk_virobiv
from Consumer, Dogovir, Virib
where Virib.id_Viroby = Dogovir.n_viroby and Consumer.id_Consumer = Dogovir.n_spozhivacha
--Роздрукувати звіт про відвантаження виробів за добу.

SELECT
    S.Nazva AS Supermarket,
    V.Imenyvannya AS Virob,
    T.Data_Vidprav AS Data_Vidprav,
    C.PIB AS Spozhivach
FROM TTN AS T
JOIN Supermarket_Virib AS SV ON T.n_dani_dost = SV.id_sup_vir
JOIN Supermarket AS S ON SV.n_supermark = S.id_Supermarket
JOIN Virib AS V ON SV.n_viroby = V.id_Viroby
JOIN Consumer AS C ON T.n_dogovir = C.id_Consumer
WHERE T.Data_Vidprav = '2023-01-24';

--Роздрукувати звіт про відвантаження виробів за місяць.

SELECT
    S.Nazva AS Supermarket,
    V.Imenyvannya AS Virob,
    T.Data_Vidprav AS Data_Vidprav,
    C.PIB AS Spozhivach
FROM TTN AS T
JOIN Supermarket_Virib AS SV ON T.n_dani_dost = SV.id_sup_vir
JOIN Supermarket AS S ON SV.n_supermark = S.id_Supermarket
JOIN Virib AS V ON SV.n_viroby = V.id_Viroby
JOIN Consumer AS C ON T.n_dogovir = C.id_Consumer
WHERE MONTH(T.Data_Vidprav) = 3; --Можна змінювати місяць на бажаний

--Сформувати список виробів для окремого споживача.

select Consumer.PIB, Virib.Imenyvannya, Dogovir.Kilk_virobiv
from Consumer join Dogovir on id_Consumer = n_spozhivacha
join Virib on n_viroby = id_Viroby

```

Рис. 6 Виконані індивідуальні завдання

4. Створити наступні статистичні процедури:

процедуру, що визначає кількість рядків в таблицях БД і заносить результат в нову таблицю.

процедуру, що визначає кількість полів в таблицях БД і заносить результат в нову таблицю.

процедуру, що визначає для кожного поля таблиці, кількість значень, що не повторюються.

```

--Завдання №4.1

CREATE PROCEDURE CalculateRowCounts
AS
BEGIN
    -- Створення тимчасової таблиці для зберігання результатів
    CREATE TABLE #RowCountResults (TableName NVARCHAR(128), RowCountValue INT)

    -- Отримання списку всіх таблиць у базі даних
    DECLARE @TableName NVARCHAR(128)
    DECLARE table_cursor CURSOR FOR
    SELECT name FROM sys.tables

    -- Для кожної таблиці отримуємо кількість рядків та записуємо в тимчасову таблицю
    OPEN table_cursor
    FETCH NEXT FROM table_cursor INTO @TableName
    WHILE @@FETCH_STATUS = 0
    BEGIN
        DECLARE @DynamicSQL NVARCHAR(MAX)
        SET @DynamicSQL = 'INSERT INTO #RowCountResults (TableName, RowCountValue) ' +
            'SELECT ''' + @TableName + ''', COUNT(*) FROM ' + @TableName
        EXEC sp_executesql @DynamicSQL

        FETCH NEXT FROM table_cursor INTO @TableName
    END

    CLOSE table_cursor
    DEALLOCATE table_cursor

    -- Виведення результатів
    SELECT * FROM #RowCountResults

    -- Видалення тимчасової таблиці
    DROP TABLE #RowCountResults
END

exec CalculateRowCounts

```

Рис. 7 Виконане завдання 4.1

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.5.000 – Лр.1	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

--Завдання №4.2
-- Створимо зброєну процедуру для рахунку кількості стовпців у таблицях
CREATE PROCEDURE CalculateColumnCounts
AS
BEGIN
    -- Створимо тимчасову таблицю для зберігання результатів
    CREATE TABLE #ColumnCountResults (
        TableName NVARCHAR(128),
        ColumnCount INT
    )

    -- Отримаємо список всіх таблиць у базі даних
    DECLARE @TableName NVARCHAR(128)
    DECLARE @SQL NVARCHAR(MAX)

    DECLARE table_cursor CURSOR FOR
    SELECT name FROM sys.tables

    -- Очистимо тимчасову таблицю перед початком
    TRUNCATE TABLE #ColumnCountResults

    -- Для кожної таблиці отримуємо кількість стовпців та записуємо в тимчасову таблицю
    OPEN table_cursor
    FETCH NEXT FROM table_cursor INTO @TableName
    WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @SQL = 'INSERT INTO #ColumnCountResults (TableName, ColumnCount) ' +
            'SELECT ''' + @TableName + ''', COUNT(*) ' +
            'FROM INFORMATION_SCHEMA.COLUMNS ' +
            'WHERE TABLE_NAME = ''' + @TableName + ''''
        EXEC sp_executesql @SQL

        FETCH NEXT FROM table_cursor INTO @TableName
    END

    CLOSE table_cursor
    DEALLOCATE table_cursor

    -- Виведення результатів
    SELECT * FROM #ColumnCountResults

    -- Видалення тимчасової таблиці
    DROP TABLE #ColumnCountResults
END

exec CalculateColumnCounts

```

Рис. 8 Виконане завдання 4.2

```

--Завдання №4.3
-- Створимо статистичну процедуру для визначення кількості унікальних значень для кожного поля в таблиці
CREATE PROCEDURE CalculateUniqueValueCounts
AS
BEGIN
    -- Отримаємо список всіх стовпців у таблиці
    DECLARE @TableName NVARCHAR(128) = 'Consumer'; -- Замінити на потрібну назву таблиці
    DECLARE @ColumnName NVARCHAR(128)
    DECLARE @SQL NVARCHAR(MAX)

    -- Створимо тимчасову таблицю для збереження результатів
    CREATE TABLE #UniqueValueCounts (
        TableName NVARCHAR(128),
        ColumnName NVARCHAR(128),
        UniqueValueCount INT
    )

    -- Очистимо тимчасову таблицю перед початком
    TRUNCATE TABLE #UniqueValueCounts

    -- Отримаємо список стовпців зі значеннями та кількістю унікальних значень
    DECLARE column_cursor CURSOR FOR
    SELECT COLUMN_NAME
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE TABLE_NAME = @TableName

    OPEN column_cursor
    FETCH NEXT FROM column_cursor INTO @ColumnName

    WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @SQL = 'INSERT INTO #UniqueValueCounts (TableName, ColumnName, UniqueValueCount) ' +
            'SELECT ''' + @TableName + ''', ''' + @ColumnName + ''', COUNT(DISTINCT ' + @ColumnName + ') ' +
            'FROM ' + @TableName
        EXEC sp_executesql @SQL

        FETCH NEXT FROM column_cursor INTO @ColumnName
    END

    CLOSE column_cursor
    DEALLOCATE column_cursor

    -- Виведення результатів
    SELECT * FROM #UniqueValueCounts

    -- Видалення тимчасової таблиці
    DROP TABLE #UniqueValueCounts
END

exec CalculateUniqueValueCounts

```

Рис. 9 Виконане завдання 4.3

5. Створити наступні тригери:

а. Для операції оновлення таблиці:

- ☐ Тригер, що оновлює одночасно дані в двох таблицях
- ☐ Тригер, що при оновленні відхиляє зміни, якщо є зв'язані дані в іншій таблиці

б. Для операції знищення даних з таблиці:

- ☐ Тригер, що знищує зв'язані дані одночасно в двох таблицях
- ☐ Тригер, що при знищенні перевіряє наявність в іншій таблиці даних, що відповідають заданій умові і відхиляє знищення даних

с. Для операції вставки даних:

- ☐ Створити тригер, що реалізує вставку даних і зміну кількості рядків в таблиці (дані про кількість рядків в таблицях містяться в окремій таблиці).

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.5.000 – Лр.1	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

--Завдання №5
--a.1
select *
from Consumer
select *
from Dogovir
CREATE TRIGGER UpdateDataTrigger
ON Consumer
AFTER UPDATE
AS
BEGIN
    UPDATE Dogovir
    SET Nomer_Dogovory = i.PoshtIndex
    FROM Dogovir d1
    INNER JOIN inserted i ON d1.n_spozhyvacha = i.id_Consumer;
END

-- Приклад виконання
select *
from Consumer
select *
from Dogovir
Update Consumer
set PoshtIndex = 13254
where id_Consumer = 1
select *
from Consumer
select *
from Dogovir

```

Рис. 10 Виконане завдання 5.а.1

```

-- a.2
CREATE TRIGGER trg_Update_Virib ON Virib
FOR UPDATE
AS
IF EXISTS (SELECT 1 FROM inserted i JOIN Supermarket_Virib sv ON i.id_Viroby = sv.n_viroby)
BEGIN
    RAISERROR ('Не можна оновити дані, оскільки вони використовуються в іншій таблиці', 16, 1);
    ROLLBACK TRANSACTION;
    RETURN;
END;

-- example
select *
from Supermarket_Virib
select *
from Virib
Update Virib
set Price = 15.50
where id_Viroby = 1

```

Рис. 11 Виконане завдання 5.а.2


```

-- b.1
CREATE TRIGGER trg_Delete_Virib ON Virib
FOR DELETE
AS
DELETE FROM Supermarket_Virib WHERE n_viroby IN (SELECT id_Viroby FROM deleted);
DELETE FROM Dogovir WHERE n_viroby IN (SELECT id_Viroby FROM deleted);

--example
select *
from Virib

Delete from Virib
where id_Viroby = 2

-- b.2
CREATE TRIGGER trg_Delete_Consumer ON Consumer
FOR DELETE
AS
IF EXISTS (SELECT 1 FROM deleted d JOIN Dogovir dg ON d.id_Consumer = dg.n_spozivacha)
BEGIN
    RAISERROR ('Не можна видалити дані, оскільки вони використовуються в іншій таблиці', 16, 1);
    ROLLBACK TRANSACTION;
    RETURN;
END;

--example
select *
from Consumer
select *
from Dogovir

-- Відключити обмеження зовн ключа, для працездатності тригера
ALTER TABLE Dogovir NOCHECK CONSTRAINT FK__Dogovir__n_spoz__5629CD9C

-- Спробувати видалити запис (Тригер працює)
DELETE FROM Consumer WHERE id_Consumer = 1

-- Включити обмеження зовн ключа (Після перевірки тригера вмикнути обмеження)
ALTER TABLE Dogovir CHECK CONSTRAINT FK__Dogovir__n_spoz__5629CD9C

```

Рис. 12 Виконане завдання 5.b.1 та 5.b.2

```

-- c.1
CREATE TABLE RowCounts (
    TableName varchar(128),
    RowCnt int
)

CREATE TRIGGER trg_Update_RowCounts ON Virib
AFTER INSERT, DELETE, UPDATE
AS
BEGIN
    -- Перевірка наявності запису для таблиці 'Virib' в таблиці 'RowCounts'
    IF EXISTS (SELECT 1 FROM RowCounts WHERE TableName = 'Virib')
        -- Якщо запис існує, оновлюємо кількість рядків
        UPDATE RowCounts SET RowCnt = (SELECT COUNT(*) FROM Virib) WHERE TableName = 'Virib'
    ELSE
        -- Якщо запису не існує, створюємо новий
        INSERT INTO RowCounts (TableName, RowCnt) VALUES ('Virib', (SELECT COUNT(*) FROM Virib))
END

drop TRIGGER trg_Insert_Virib
drop table RowCounts

--example
select *
from RowCounts

select *
from Virib

delete
from Virib
where Kod_viroby = 4

INSERT INTO Virib (Kod_viroby, Imenyvannya, Price)
VALUES
(4, 'Торт', 50.00),
(5, 'Пиріг', 25.50);

```

Рис. 13 Виконане завдання 5.c.1

Висновок: дослідив засоби реалізації БД та SQL в середовищі MSSQL Server.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.5.000 – Лр.1	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		