-- Создание базы данных "Аптека"

CREATE DATABASE Аптека;

GO

-- Использование базы данных "Аптека"

USE Аптека;

GO

-- Создание таблицы "Препараты"

CREATE TABLE Препараты (

id\_препарата INT PRIMARY KEY,

Название VARCHAR(100) NOT NULL,

Производитель VARCHAR(100) NOT NULL,

Страна\_производства VARCHAR(50) NOT NULL

);

GO

-- Создание таблицы "Аптеки"

CREATE TABLE Аптеки (

id\_аптеки INT PRIMARY KEY,

Название VARCHAR(100) NOT NULL,

Адрес VARCHAR(255) NOT NULL,

Телефон VARCHAR(20) NOT NULL

);

GO

-- Создание таблицы "Закупки"

CREATE TABLE Закупки (

id\_закупки INT PRIMARY KEY,

id\_аптеки INT,

Дата DATETIME NOT NULL,

FOREIGN KEY (id\_аптеки) REFERENCES Аптеки(id\_аптеки)

);

GO

-- Создание таблицы "Закупки\_Препараты"

CREATE TABLE Закупки\_Препараты (

id\_закупки INT,

id\_препарата INT,

Количество INT NOT NULL,

FOREIGN KEY (id\_закупки) REFERENCES Закупки(id\_закупки),

FOREIGN KEY (id\_препарата) REFERENCES Препараты(id\_препарата)

);

GO

use Аптека

go

-- Заполнение таблицы "Препараты"

INSERT INTO Препараты (id\_препарата, Название, Производитель, Страна\_производства)

VALUES

(1, 'Анальгин', 'Фармкомпания', 'Россия'),

(2, 'Аспирин', 'Байер', 'Германия'),

(3, 'Ибупрофен', 'Фармалайн', 'Россия'),

(4, 'Но-шпа', 'Санофи Авентис', 'Франция'),

(5, 'Цитрамон', 'Хиноин', 'Венгрия');

-- Заполнение таблицы "Аптеки"

INSERT INTO Аптеки (id\_аптеки, Название, Адрес, Телефон)

VALUES

(1, 'Аптека №1', 'ул. Центральная 1', '123-456-7890'),

(2, 'Аптека №2', 'ул. Первомайская 2', '987-654-3210'),

(3, 'Аптека "Здоровье"', 'пр. Победы 10', '555-555-5555');

-- Заполнение таблицы "Закупки"

INSERT INTO Закупки (id\_закупки, id\_аптеки, Дата)

VALUES

(1, 1, '2023-05-01 10:00:00'),

(2, 2, '2023-05-05 15:30:00'),

(3, 3, '2023-05-10 12:15:00');

-- Заполнение таблицы "Закупки\_Препараты"

INSERT INTO Закупки\_Препараты (id\_закупки, id\_препарата, Количество)

VALUES

(1, 1, 100),

(1, 2, 200),

(2, 2, 150),

(2, 3, 300),

(3, 4, 50),

(3, 5, 100),

(3, 1, 150);

use Аптека

go

CREATE PROCEDURE ДобавитьПрепарат

@Название VARCHAR(100),

@Производитель VARCHAR(100),

@Страна\_производства VARCHAR(50)

AS

BEGIN

INSERT INTO Препараты (Название, Производитель, Страна\_производства)

VALUES (@Название, @Производитель, @Страна\_производства);

END;

go

CREATE PROCEDURE ПолучитьАптекуПоНазванию

@Название VARCHAR(100)

AS

BEGIN

SELECT \*

FROM Аптеки

WHERE Название = @Название;

END;

go

CREATE PROCEDURE ПолучитьЗакупкиПоДате

@Дата DATETIME

AS

BEGIN

SELECT \*

FROM Закупки

WHERE Дата = @Дата;

END;

Go

use Аптека

go

CREATE FUNCTION ПолучитьКоличествоПрепаратовВАптеке(@id\_аптеки INT)

RETURNS INT

AS

BEGIN

DECLARE @количество INT;

SELECT @количество = SUM(Количество) FROM Закупки\_Препараты

WHERE id\_закупки IN (SELECT id\_закупки FROM Закупки WHERE id\_аптеки = @id\_аптеки);

RETURN @количество;

END;

go

CREATE FUNCTION ПолучитьПроизводителяПрепарата(@id\_препарата INT)

RETURNS VARCHAR(100)

AS

BEGIN

DECLARE @производитель VARCHAR(100);

SELECT @производитель = Производитель FROM Препараты WHERE id\_препарата = @id\_препарата;

RETURN @производитель;

END;

go

CREATE FUNCTION ПолучитьСписокЗакупокПоДате(@дата DATETIME)

RETURNS TABLE

AS

RETURN

(

SELECT Закупки.id\_закупки, Аптеки.Название AS Название\_аптеки, Закупки.Дата AS Дата\_закупки

FROM Закупки

INNER JOIN Аптеки ON Закупки.id\_аптеки = Аптеки.id\_аптеки

WHERE Закупки.Дата = @дата

);

go

use Аптека

go

CREATE TRIGGER УдалениеЗакупки

ON Закупки

AFTER DELETE

AS

BEGIN

-- Логика триггера

DELETE FROM Закупки\_Препараты

WHERE id\_закупки IN (SELECT id\_закупки FROM deleted);

END;

go

CREATE TRIGGER АптекаПрепаратыAfterInsert

ON Аптеки

AFTER INSERT

AS

BEGIN

-- Логика триггера

UPDATE Препараты

SET Страна\_производства = 'Новая страна'

WHERE id\_препарата IN (SELECT id\_препарата FROM inserted);

END;

go

CREATE TRIGGER ПрепаратыBeforeUpdate

ON Препараты

INSTEAD OF UPDATE

AS

BEGIN

-- Логика триггера

IF UPDATE(Производитель) OR UPDATE(Страна\_производства)

BEGIN

RAISERROR('Изменение производителя или страны производства запрещено!', 16, 1)

ROLLBACK;

END;

ELSE

BEGIN

UPDATE p

SET p.Название = i.Название

FROM Препараты p

INNER JOIN inserted i ON p.id\_препарата = i.id\_препарата;

END;

END;

Go

use Аптека

go

CREATE VIEW Препараты\_Производители AS

SELECT p.Название AS Препарат, p.Производитель, p.Страна\_производства

FROM Препараты p;

go

CREATE VIEW Аптеки\_Закупки AS

SELECT a.Название AS Аптека, z.Дата, COUNT(zp.id\_препарата) AS Количество\_видов\_закупленных\_препаратов

FROM Аптеки a

JOIN Закупки z ON a.id\_аптеки = z.id\_аптеки

JOIN Закупки\_Препараты zp ON z.id\_закупки = zp.id\_закупки

GROUP BY a.Название, z.Дата;

go

CREATE VIEW Сводная\_Закупки\_Препараты AS

SELECT z.Дата, p.Название AS Препарат, SUM(zp.Количество) AS Общее\_количество

FROM Закупки z

JOIN Закупки\_Препараты zp ON z.id\_закупки = zp.id\_закупки

JOIN Препараты p ON zp.id\_препарата = p.id\_препарата

GROUP BY z.Дата, p.Название;

go

1. **Управление параметрами пользователей с помощью групповых политик**

Управление параметрами пользователей с помощью групповых политик является одним из ключевых аспектов в области администрирования информационных систем. Групповые политики позволяют администраторам централизованно управлять настройками и параметрами пользователей, что обеспечивает единообразие и согласованность в сети.

С помощью групповых политик можно контролировать и настраивать различные аспекты работы пользователей, включая:

1. Безопасность: Групповые политики позволяют управлять политиками паролей, требованиями к безопасности, доступом к ресурсам и другими аспектами безопасности.
2. Профили пользователей: Групповые политики позволяют настроить профили пользователей, включая настройки рабочего стола, темы оформления, настройки приложений и другие параметры, которые могут быть предварительно настроены и автоматически применены при входе пользователя в систему.
3. Приложения: Групповые политики могут использоваться для управления настройками и поведением приложений.
4. Сетевые ресурсы: Групповые политики позволяют управлять доступом пользователей к сетевым ресурсам, таким как общие папки, принтеры, сетевые диски и т.д.
5. Ограничения: Групповые политики могут использоваться для установки ограничений и правил для пользователей.