Создать базу данных **Telefon**

В БД Telefon добавьте таблицу Abonent с данными абонентов (номер абонента, фамилия и имя, дата оплаты, абонентская плата). Для поля номер абонента установить ограничение первичного ключа. Остальные поля таблицы обязательны для заполнения. А для поля абонентская плата еще добавьте ограничение значения по умолчанию - 600 руб.

Разработать процедуру (Vvod tel) для ввода данных в таблицу Abonent. Ввести 5 записей. Для проверки ввода, покажите содержимое таблицы.

Разработать представление View\_tel, которое будет выводить информацию об общем количестве абонентов.

Создать в БД роль Mobil и двух пользователей Abonent1и Abonent2, у которых пароли для входа в базу данных: Ab123 и Ab456.

Поместить созданных пользователей в роль Mobil. Для роли определите права на чтение и изменение таблицы Abonent, права на работу с представлением View tel и на запуск процедуры Vvod tel. Пользователю Abonent1 определите разрешение на выполнение копии базы данных Telefon. используя системную процедуру, проверьте правильность выданных прав пользователям.

В целях безопасности данных выполните копирование базы данных на созданное вами устройство резервного копирования. Выведите информацию на экран о созданной базе данных.

CREATE DATABASE Telefon;

GO

CREATE DATABASE Telefon;

GO

USE Telefon;

GO

drop table Абонент

CREATE TABLE Абонент (

Номер\_абонента INT PRIMARY KEY,

Фамилия NVARCHAR(50) NOT NULL,

Имя NVARCHAR(50) NOT NULL,

Дата\_оплаты DATE NOT NULL,

Абонентская\_плата DECIMAL(10, 2) NOT NULL DEFAULT 600

);

GO

-- Создание процедуры для ввода данных

Drop procedure Vvod\_tel

CREATE PROCEDURE Vvod\_tel

@Номер\_абонента INT,

@Фамилия NVARCHAR(50),

@Имя NVARCHAR(50),

@Дата\_оплаты DATE,

@Абонентская\_плата DECIMAL(10, 2) = 600

AS

BEGIN

INSERT INTO Абонент (Номер\_абонента, Фамилия, Имя, Дата\_оплаты, Абонентская\_плата)

VALUES (@Номер\_абонента, @Фамилия, @Имя, @Дата\_оплаты, @Абонентская\_плата);

END;

GO

-- Ввод 5 записей и проверка данных

EXEC Vvod\_tel 1, N'Иванов', N'Иван', '2024-09-01', 600;

EXEC Vvod\_tel 2, N'Петров', N'Петр', '2024-09-10', 650;

EXEC Vvod\_tel 3, N'Сидоров', N'Сидор', '2024-09-15', 600;

EXEC Vvod\_tel 4, N'Федоров', N'Федор', '2024-09-20', 700;

EXEC Vvod\_tel 5, N'Морозов', N'Мороз', '2024-09-25', 600;

-- Проверка содержимого таблицы

SELECT \* FROM Абонент;

GO

--Создание представления

CREATE VIEW View\_tel AS

SELECT COUNT(\*) AS Общее\_количество\_абонентов FROM Абонент;

GO

-- Проверка представления

SELECT \* FROM View\_tel;

GO

--Создание ролей и пользователей

-- Создание роли Mobil

CREATE ROLE Mobil;

GO

-- Создание пользователей

CREATE LOGIN Abonent1 WITH PASSWORD = 'Ab123';

CREATE LOGIN Abonent2 WITH PASSWORD = 'Ab456';

CREATE USER Abonent1 FOR LOGIN Abonent1;

CREATE USER Abonent2 FOR LOGIN Abonent2;

-- Назначение пользователей в роль Mobil

ALTER ROLE Mobil ADD MEMBER Abonent1;

ALTER ROLE Mobil ADD MEMBER Abonent2;

GO

--Назначение прав

-- Права для роли Mobil на таблицу Абонент

GRANT SELECT, UPDATE ON Абонент TO Mobil;

-- Права для роли Mobil на представление View\_tel

GRANT SELECT ON View\_tel TO Mobil;

-- Права для роли Mobil на выполнение процедуры Vvod\_tel

GRANT EXECUTE ON Vvod\_tel TO Mobil;

-- Дополнительные права для Abonent1 на выполнение резервного копирования

GRANT BACKUP DATABASE TO Abonent1;

GO

-- Проверка прав

EXECUTE sp\_helprotect @username = 'Abonent1';

EXECUTE sp\_helprotect @username = 'Abonent2';

-- Создание устройства резервного копирования

EXEC sp\_addumpdevice 'disk', 'TelefonBackup', 'C:\Backup\Telefon.bak';

-- Выполнение резервного копирования

BACKUP DATABASE Telefon TO TelefonBackup;

GO

-- Информация о базе данных

EXEC sp\_helpdb 'Telefon';

GO

**SHOP**

Создать базу данных Shop. используя оператор select, сделайте копию таблицы Order Details из бд northwind в бд Shop под именем ORDDetails1.

Разработать процедуру Del\_Shop для удаления новой таблицы ORDDetails1 те записи, у которых самая наименьшая и наибольшая цена.

Разработать процедуру Upd\_Shop, увеличивающую цену (поле UnitPrice) на 100 руб записям, у которых значение поля OrderID равно 10265

Разработать преставление View\_Shop, которое позволит получить информацию о записях, в которых значение поля OrderID равно 10250 и 10255

Создать в БД роль RoleShop и двух пользователей Shop1 и Shop2, у которых будут пароли для входа в базу данных: Shop234 и Shop235.

Поместить созданных пользователей в роль RoleShop. Для роли определите права на запуск процедуры Del\_Shop. Пользователю Shop2 определите разрешение на чтение и изменение данных таблицы ORDDetails1. используя системную процедуру, проверьте правильность выданных прав пользователям.

-- Создать базу данных Shop

CREATE DATABASE Shop;

GO

-- Использовать базу данных Shop

USE Shop;

GO

-- Копирование таблицы Order Details из базы данных Northwind в Shop под именем ORDDetails1

SELECT \*

INTO ORDDetails1

FROM Northwind.dbo.[Order Details];

GO

-- Создание процедуры Del\_Shop для удаления записей с наименьшей и наибольшей ценой

CREATE PROCEDURE Del\_Shop AS

BEGIN

DELETE FROM ORDDetails1

WHERE UnitPrice = (SELECT MIN(UnitPrice) FROM ORDDetails1)

OR UnitPrice = (SELECT MAX(UnitPrice) FROM ORDDetails1);

END;

GO

-- Создание процедуры Upd\_Shop для увеличения цены на 100 руб

CREATE PROCEDURE Upd\_Shop AS

BEGIN

UPDATE ORDDetails1

SET UnitPrice = UnitPrice + 100

WHERE OrderID = 10265;

END;

GO

-- Создание представления View\_Shop

CREATE VIEW View\_Shop AS

SELECT \*

FROM ORDDetails1

WHERE OrderID IN (10250, 10255);

GO

-- Проверка представления

SELECT \* FROM View\_Shop;

GO

-- Создание роли RoleShop

CREATE ROLE RoleShop;

GO

-- Создание пользователей Shop1 и Shop2

CREATE LOGIN Shop1 WITH PASSWORD = 'Shop234';

CREATE LOGIN Shop2 WITH PASSWORD = 'Shop235';

CREATE USER Shop1 FOR LOGIN Shop1;

CREATE USER Shop2 FOR LOGIN Shop2;

-- Назначение пользователей в роль RoleShop

ALTER ROLE RoleShop ADD MEMBER Shop1;

ALTER ROLE RoleShop ADD MEMBER Shop2;

GO

-- Права для роли RoleShop на выполнение процедуры Del\_Shop

GRANT EXECUTE ON Del\_Shop TO RoleShop;

-- Права для пользователя Shop2 на чтение и изменение данных таблицы ORDDetails1

GRANT SELECT, UPDATE ON ORDDetails1 TO Shop2;

GO

EXECUTE sp\_helprotect @username = 'Shop1';

EXECUTE sp\_helprotect @username = 'Shop2';

Sphere

Создать базу данных Sphere.

Создайте две таблицы Member и Adult.

В member: member\_no, lastname, firstname, middleinitial, photograph.

В Adult: member\_no, street, city, state, phone\_no.

Нужно установить связь между таблицами. типы данных определить согласно тем данным, которые там будут храниться.

используя оператор ввода, введите в таблицы три записи.

Разработать процедуру Upd\_Sphere на обновление в таблице Adult значений в поле phone\_no.

Разработать преставление View\_Sphere, которое позволит получить информацию о конкретном участнике. В представление входят поля member\_no, lastname, city, phone\_no.

Создать в БД роль Sphere1 и двух пользователей Sph1 и Sph2, у которых будут пароли для входа в базу данных: Sph333 и Sph444.

use master

CREATE DATABASE Sphere;

GO

USE Sphere;

GO

CREATE TABLE Member (

member\_no INT PRIMARY KEY,

lastname NVARCHAR(50) NOT NULL,

firstname NVARCHAR(50) NOT NULL,

middleinitial NCHAR(1),

photograph VARBINARY(MAX)

);

CREATE TABLE Adult (

member\_no INT PRIMARY KEY,

street NVARCHAR(100),

city NVARCHAR(50),

state NVARCHAR(50),

phone\_no NVARCHAR(20),

CONSTRAINT FK\_Adult\_Member FOREIGN KEY (member\_no) REFERENCES Member(member\_no)

);

INSERT INTO Member (member\_no, lastname, firstname, middleinitial)

VALUES

(1, 'Иванов', 'Иван', 'И'),

(2, 'Петров', 'Петр', 'П'),

(3, 'Сидоров', 'Сидор', 'С');

INSERT INTO Adult (member\_no, street, city, state, phone\_no)

VALUES

(1, 'ул. Ленина, 10', 'Москва', 'Московская обл.', '+7(123)456-78-90'),

(2, 'ул. Пушкина, 20', 'Санкт-Петербург', 'Ленинградская обл.', '+7(987)654-32-10'),

(3, 'ул. Гагарина, 30', 'Новосибирск', 'Новосибирская обл.', '+7(555)123-45-67');

go

-- Создание процедуры Upd\_Sphere для обновления phone\_no

CREATE PROCEDURE Upd\_Sphere

@member\_no INT,

@new\_phone\_no NVARCHAR(20)

AS

BEGIN

UPDATE Adult

SET phone\_no = @new\_phone\_no

WHERE member\_no = @member\_no;

END

go

-- Создание представления View\_Sphere

CREATE VIEW View\_Sphere AS

SELECT m.member\_no, m.lastname, a.city, a.phone\_no

FROM Member m

JOIN Adult a ON m.member\_no = a.member\_no;

GO

-- Проверка представления

SELECT \* FROM View\_Sphere;

GO

-- Создание пользователей Sph1 и Sph2

CREATE LOGIN Sph1 WITH PASSWORD = 'Sph333';

CREATE LOGIN Sph2 WITH PASSWORD = 'Sph444';

CREATE USER Sph1 FOR LOGIN Sph1;

CREATE USER Sph2 FOR LOGIN Sph2;

-- Назначение пользователей в роль Sphere1

ALTER ROLE Sphere1 ADD MEMBER Sph1;

ALTER ROLE Sphere1 ADD MEMBER Sph2;

GO

-- Права для роли Sphere1 на выполнение процедуры Upd\_Sphere

GRANT EXECUTE ON Upd\_Sphere TO Sphere1;

-- Права для роли Sphere1 на чтение представления View\_Sphere

GRANT SELECT ON View\_Sphere TO Sphere1;

Go

MyStud

Создать базу данных MyStud со следующими размерами основного файла - 5 МБ, журнала транзакций - 2 МБ. Для двух файлов указать шаг прироста - 1 Мб. Максимальный размер основного файла - 10 Мб, журнала - 5 Мб. Проверить существует ли данная БД.

В БД MyStud добавьте таблицу Anketa с анкетными данными студентов (шифр, фамилия, имя, год рождения, город, возраст). Для поля шифр установить ограничение первичного ключа. Остальные поля таблицы обязательны для заполнения. А для поля город еще добавьте ограничение значения по умолчанию - СПБ.

Разработать процедуру (Vvod Stud) для ввода данных в таблицу Anketa. Ввести 5 записей. Для проверки ввода, покажите содержимое таблицы.

Разработать представление View\_Stud, которое будет выводить информацию обо всех студентах, рожденных в период с 2000 по 2002 год.

Используя оператор языка Transact SQL, увеличьте в два раза значение в поле возраст студенту с указанной фамилией. Проверте результат.

Создать в БД роль RoleStud и двух пользователей Stud1 и Stud2, у которых пароли для входа в базу данных: Stud777 и Stud666.

Поместить созданных пользователей в роль RoleStud. Для роли определите права на чтение, удаление и изменение таблицы Anketa. Пользователю Stud2, определите разрешение на выполнение копии базы данных MyStud. используя системную процедуру, проверьте правильность выданных прав пользователям.

В целях безопасности данных выполните копирование базы данных на созданное вами устройство резервног копирования. Выведите информацию на экран о созданной базе данных.

USE master;

GO

-- Создание базы данных

CREATE DATABASE MyStud

USE MyStud;

GO

CREATE TABLE Anketa

(

Шифр INT PRIMARY KEY,

Фамилия NVARCHAR(50) NOT NULL,

Имя NVARCHAR(50) NOT NULL,

ГодРождения INT NOT NULL,

Город NVARCHAR(50) NOT NULL DEFAULT 'СПБ',

Возраст INT NOT NULL

);

GO

select \* from Anketa

go

CREATE PROCEDURE Vvod\_Stud

@Шифр INT,

@Фамилия NVARCHAR(50),

@Имя NVARCHAR(50),

@ГодРождения INT,

@Город NVARCHAR(50),

@Возраст INT

AS

BEGIN

INSERT INTO Anketa (Шифр, Фамилия, Имя, ГодРождения, Город, Возраст)

VALUES (@Шифр, @Фамилия, @Имя, @ГодРождения, @Город, @Возраст);

END

GO

-- Ввод 5 записей

EXEC Vvod\_Stud 1, 'Иванов', 'Иван', 2000, 'Москва', 23;

EXEC Vvod\_Stud 2, 'Петров', 'Петр', 2001, 'СПБ', 22;

EXEC Vvod\_Stud 3, 'Сидорова', 'Мария', 2002, 'Казань', 21;

EXEC Vvod\_Stud 4, 'Козлов', 'Алексей', 1999, 'СПБ', 24;

EXEC Vvod\_Stud 5, 'Смирнова', 'Елена', 2003, 'Новосибирск', 20;

go

CREATE VIEW View\_Stud AS

SELECT \*

FROM Anketa

WHERE ГодРождения between 2000 and 2002;

GO

-- Проверка представления

SELECT \* FROM View\_Stud

go

UPDATE Anketa

SET Возраст = Возраст \* 2

WHERE Фамилия = 'Иванов';

-- Проверка результата

SELECT \* FROM Anketa WHERE Фамилия = 'Иванов';

go

-- Создание роли

CREATE ROLE RoleStud;

-- Создание пользователей

CREATE LOGIN Stud1 WITH PASSWORD = 'Stud777';

CREATE USER Stud1 FOR LOGIN Stud1;

CREATE LOGIN Stud2 WITH PASSWORD = 'Stud666';

CREATE USER Stud2 FOR LOGIN Stud2;

-- Добавление пользователей в роль

ALTER ROLE RoleStud ADD MEMBER Stud1;

ALTER ROLE RoleStud ADD MEMBER Stud2;

-- Назначение прав для роли

GRANT SELECT, DELETE, UPDATE ON Anketa TO RoleStud;

-- Назначение права на создание резервной копии для Stud2

GRANT BACKUP DATABASE TO Stud2;

sp\_helpuser Stud1

sp\_helprotect