CREATE DATABASE Химчистка

GO

USE Химчистка;

GO

-- Создание таблицы "Клиенты"

CREATE TABLE Клиенты (

id\_клиента INT PRIMARY KEY IDENTITY(1, 1),

Фамилия VARCHAR(100) NOT NULL,

Имя VARCHAR(100) NOT NULL,

Телефон VARCHAR(20) NOT NULL,

Адрес VARCHAR(200) NOT NULL,

Email VARCHAR(100)

);

GO

-- Создание таблицы "Сотрудники"

CREATE TABLE Сотрудники (

id\_сотрудника INT PRIMARY KEY IDENTITY(1, 1),

Фамилия VARCHAR(100) NOT NULL,

Имя VARCHAR(100) NOT NULL,

Должность VARCHAR(100) NOT NULL,

Зарплата DECIMAL(10, 2) NOT NULL

);

GO

-- Создание таблицы "Заказы"

CREATE TABLE Заказы (

id\_заказа INT PRIMARY KEY IDENTITY(1, 1),

id\_клиента INT,

id\_сотрудника INT,

Дата\_приема DATE NOT NULL,

Дата\_выдачи DATE,

FOREIGN KEY (id\_клиента) REFERENCES Клиенты(id\_клиента),

FOREIGN KEY (id\_сотрудника) REFERENCES Сотрудники(id\_сотрудника),

);

GO

-- Создание таблицы "Вещи"

CREATE TABLE Вещи (

id\_вещи INT PRIMARY KEY IDENTITY(1, 1),

Наименование VARCHAR(100) NOT NULL,

Описание VARCHAR(200),

Стоимость DECIMAL(10, 2) NOT NULL

);

GO

-- Создание таблицы "Заказанные\_вещи"

CREATE TABLE Заказанные\_вещи (

id\_заказа INT,

id\_вещи INT,

Количество INT NOT NULL,

FOREIGN KEY (id\_заказа) REFERENCES Заказы(id\_заказа),

FOREIGN KEY (id\_вещи) REFERENCES Вещи(id\_вещи)

);

GO

USE Химчистка;

GO

-- Заполнение таблицы "Клиенты"

INSERT INTO Клиенты (Фамилия, Имя, Телефон, Адрес, Email)

VALUES

('Иванов', 'Алексей', '1234567890', 'ул. Ленина, 1', 'ivanov@example.com'),

('Петров', 'Иван', '9876543210', 'ул. Пушкина, 2', 'petrov@example.com'),

('Сидорова', 'Елена', '555444333', 'ул. Советская, 3', 'sidorova@example.com'),

('Козлова', 'Анна', '111222333', 'ул. Гагарина, 4', 'kozlova@example.com');

-- Заполнение таблицы "Сотрудники"

INSERT INTO Сотрудники (Фамилия, Имя, Должность, Зарплата)

VALUES

('Смирнов', 'Александр', 'Клининг-мастер', 20000),

('Козлов', 'Андрей', 'Глажильщик', 25000),

('Петрова', 'Ольга', 'Швея', 30000),

('Иванова', 'Мария', 'Швея', 35000);

-- Заполнение таблицы "Заказы"

INSERT INTO Заказы (id\_клиента, id\_сотрудника, Дата\_приема, Дата\_выдачи)

VALUES

(1, 1, '2023-05-01', '2023-05-05'),

(2, 2, '2023-05-02', '2023-05-06'),

(3, 3, '2023-05-03', '2023-05-07'),

(4, 4, '2023-05-04', '2023-05-08');

-- Заполнение таблицы "Вещи"

INSERT INTO Вещи (Наименование, Описание, Стоимость)

VALUES

('Пальто', 'Черное, размер 42', 1500),

('Брюки', 'Синие джинсы, размер 32', 800),

('Платье', 'Красное, размер 38', 2000),

('Рубашка', 'Белая, размер 40', 600),

('Пиджак', 'Серый, размер 44', 1200);

-- Заполнение таблицы "Заказанные\_вещи"

INSERT INTO Заказанные\_вещи (id\_заказа, id\_вещи, Количество)

VALUES

(1, 1, 1),

(1, 2, 2),

(2, 3, 1),

(2, 4, 2),

(3, 5, 3),

(4, 1, 2),

(4, 3, 1);

CREATE VIEW Список\_Заказов AS

SELECT Заказы.id\_заказа, Клиенты.Фамилия, Клиенты.Имя, Заказы.Дата\_приема, Заказы.Дата\_выдачи, Вещи.Наименование, Заказанные\_вещи.Количество

FROM Заказы

JOIN Клиенты ON Заказы.id\_клиента = Клиенты.id\_клиента

JOIN Заказанные\_вещи ON Заказы.id\_заказа = Заказанные\_вещи.id\_заказа

JOIN Вещи ON Заказанные\_вещи.id\_вещи = Вещи.id\_вещи;

CREATE VIEW Список\_Сотрудников\_Заказов AS

SELECT Сотрудники.id\_сотрудника, Сотрудники.Фамилия, Сотрудники.Имя, Заказы.id\_заказа, Заказы.Дата\_приема, Заказы.Дата\_выдачи

FROM Сотрудники

JOIN Заказы ON Сотрудники.id\_сотрудника = Заказы.id\_сотрудника;

CREATE VIEW Стоимость\_Заказов AS

SELECT Заказы.id\_заказа, SUM(Вещи.Стоимость \* Заказанные\_вещи.Количество) AS Общая\_стоимость

FROM Заказы

JOIN Заказанные\_вещи ON Заказы.id\_заказа = Заказанные\_вещи.id\_заказа

JOIN Вещи ON Заказанные\_вещи.id\_вещи = Вещи.id\_вещи

GROUP BY Заказы.id\_заказа;

CREATE PROCEDURE Добавить\_Заказ

@id\_клиента INT,

@id\_сотрудника INT,

@Дата\_приема DATE,

@Дата\_выдачи DATE

AS

BEGIN

INSERT INTO Заказы (id\_клиента, id\_сотрудника, Дата\_приема, Дата\_выдачи)

VALUES (@id\_клиента, @id\_сотрудника, @Дата\_приема, @Дата\_выдачи);

END;

CREATE PROCEDURE Получить\_Список\_Заказов\_По\_Клиенту

@id\_клиента INT

AS

BEGIN

SELECT \*

FROM Заказы

WHERE id\_клиента = @id\_клиента;

END;

CREATE PROCEDURE Удалить\_Заказ

@id\_заказа INT

AS

BEGIN

DELETE FROM Заказы

WHERE id\_заказа = @id\_заказа;

END;

CREATE TRIGGER Автоматическое\_Обновление\_Дата\_Выдачи

ON Заказы

AFTER INSERT

AS

BEGIN

UPDATE Заказы

SET Дата\_выдачи = DATEADD(DAY, 3, Дата\_приема)

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

END;

GO

CREATE TRIGGER Автоматическое\_Удаление\_Клиента

ON Клиенты

AFTER DELETE

AS

BEGIN

DELETE FROM Заказы

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

END;

GO

CREATE TRIGGER Автоматическое\_Удаление\_Заказанных\_Вещей

ON Заказы

AFTER DELETE

AS

BEGIN

DELETE FROM Заказанные\_вещи

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

END;

GO

CREATE FUNCTION Получить\_Общую\_Стоимость\_Заказа

(

@id\_заказа INT

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @общая\_стоимость DECIMAL(10, 2);

SELECT @общая\_стоимость = SUM(Заказанные\_вещи.Количество \* Вещи.Стоимость)

FROM Заказанные\_вещи

INNER JOIN Вещи ON Заказанные\_вещи.id\_вещи = Вещи.id\_вещи

WHERE Заказанные\_вещи.id\_заказа = @id\_заказа;

RETURN @общая\_стоимость;

END;

CREATE FUNCTION Получить\_Количество\_Заказанных\_Вещей\_По\_Заказу

(

@id\_заказа INT

)

RETURNS INT

AS

BEGIN

DECLARE @количество INT;

SELECT @количество = SUM(Количество)

FROM Заказанные\_вещи

WHERE id\_заказа = @id\_заказа;

RETURN @количество;

END;

CREATE FUNCTION Получить\_Список\_Клиентов\_С\_Заказами()

RETURNS TABLE

AS

RETURN

(

SELECT Клиенты.\*, COUNT(Заказы.id\_заказа) AS Количество\_Заказов

FROM Клиенты

LEFT JOIN Заказы ON Клиенты.id\_клиента = Заказы.id\_клиента

GROUP BY Клиенты.id\_клиента, Клиенты.Фамилия, Клиенты.Имя, Клиенты.Телефон, Клиенты.Адрес, Клиенты.Email

);

1. **Экспорт данных базы в документы пользователя**

Экспорт данных из базы данных в документы пользователя — это процесс преобразования данных, хранящихся в базе данных, в формат, который может быть удобно представлен и использован в документах или файловых форматах. Это может включать экспорт данных в текстовые файлы, электронные таблицы, PDF-документы или другие форматы.

Вот основные шаги для экспорта данных из базы данных в документы пользователя:

1. Определение формата экспорта: Сначала необходимо определить, в каком формате пользователь предпочтет получить экспортированные данные.

2. Выбор данных для экспорта: определите, какие данные из базы данных должны быть экспортированы.

3. Использование средств экспорта: В большинстве СУБД существуют инструменты и функции для экспорта данных. Это может быть встроенная функция СУБД или сторонние инструменты для упрощения процесса экспорта.

4. Настройка параметров экспорта: в некоторых случаях могут быть доступны параметры, которые позволяют настроить способ экспорта данных. Например, вы можете определить разделитель полей в CSV-файле или выбрать определенные столбцы для экспорта в электронную таблицу.

5. Выполнение экспорта: запустите процесс экспорта, используя выбранные инструменты или команды. База данных будет обработана, и данные будут преобразованы в выбранный формат экспорта.