-- Создание базы данных "Прокат автомобилей"

CREATE DATABASE ПрокатАвтомобилей;

USE ПрокатАвтомобилей;

-- Создание таблицы "Автомобили"

CREATE TABLE Автомобили (

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

Марка VARCHAR(100) NOT NULL,

Модель VARCHAR(100) NOT NULL,

Год\_выпуска INT,

Цвет VARCHAR(50),

Регистрационный\_номер VARCHAR(20) UNIQUE

);

-- Создание таблицы "Клиенты"

CREATE TABLE Клиенты (

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

Имя VARCHAR(100) NOT NULL,

Фамилия VARCHAR(100) NOT NULL,

Адрес VARCHAR(100),

Телефон VARCHAR(15),

Email VARCHAR(50)

);

-- Создание таблицы "Сотрудники"

CREATE TABLE Сотрудники (

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

Имя VARCHAR(100) NOT NULL,

Фамилия VARCHAR(100) NOT NULL,

Должность VARCHAR(50) NOT NULL,

Зарплата INT

);

-- Создание таблицы "Тарифы"

CREATE TABLE Тарифы (

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

Название VARCHAR(50) NOT NULL,

Цена INT

);

-- Создание таблицы "Прокаты"

CREATE TABLE Прокаты (

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

id\_автомобиля INT,

id\_тарифа INT,

id\_клиента INT,

id\_сотрудника INT,

Дата\_начала DATE,

Дата\_окончания DATE,

FOREIGN KEY (id\_автомобиля) REFERENCES Автомобили (id\_автомобиля),

FOREIGN KEY (id\_тарифа) REFERENCES Тарифы (id\_тарифа),

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

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

);

-- Заполнение таблицы "Автомобили"

INSERT INTO Автомобили (Марка, Модель, Год\_выпуска, Цвет, Регистрационный\_номер)

VALUES

('Toyota', 'Camry', 2019, 'Серебристый', 'A123BC'),

('Honda', 'Civic', 2020, 'Черный', 'B456DE'),

('BMW', 'X5', 2018, 'Белый', 'C789FG');

-- Заполнение таблицы "Клиенты"

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

VALUES

('Иван', 'Иванов', 'Москва', '1234567890', 'ivanov@example.com'),

('Петр', 'Петров', 'Санкт-Петербург', '9876543210', 'petrov@example.com'),

('Анна', 'Сидорова', 'Казань', '5555555555', 'sidorova@example.com');

-- Заполнение таблицы "Сотрудники"

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

VALUES

('Алексей', 'Смирнов', 'Менеджер', 50000),

('Елена', 'Иванова', 'Администратор', 40000),

('Дмитрий', 'Петров', 'Водитель', 45000);

-- Заполнение таблицы "Тарифы"

INSERT INTO Тарифы (Название, Цена)

VALUES

('Стандарт', 3000),

('Премиум', 5000),

('Эконом', 2000);

-- Заполнение таблицы "Прокаты"

INSERT INTO Прокаты (id\_автомобиля, id\_тарифа, id\_клиента, id\_сотрудника, Дата\_начала, Дата\_окончания)

VALUES

(1, 1, 2, 3, '2023-05-01', '2023-05-05'),

(2, 3, 1, 2, '2023-06-10', '2023-06-15'),

(3, 2, 3, 1, '2023-07-20', '2023-07-25');

-- Представление "Список автомобилей в прокате"

CREATE VIEW ВПрокате AS

SELECT a.id\_автомобиля, a.Марка, a.Модель, a.Год\_выпуска, a.Цвет, a.Регистрационный\_номер,

p.Дата\_начала, p.Дата\_окончания

FROM Автомобили a

JOIN Прокаты p ON a.id\_автомобиля = p.id\_автомобиля;

-- Представление "Список клиентов и их прокаты"

CREATE VIEW КлиентыИПрокаты AS

SELECT k.id\_клиента, k.Имя, k.Фамилия, k.Адрес, k.Телефон, k.Email,

p.id\_проката, p.Дата\_начала, p.Дата\_окончания

FROM Клиенты k

JOIN Прокаты p ON k.id\_клиента = p.id\_клиента;

-- Представление "Список сотрудников и их прокаты"

CREATE VIEW СотрудникиИПрокаты AS

SELECT s.id\_сотрудника, s.Имя, s.Фамилия, s.Должность, s.Зарплата,

p.id\_проката, p.Дата\_начала, p.Дата\_окончания

FROM Сотрудники s

JOIN Прокаты p ON s.id\_сотрудника = p.id\_сотрудника;

--Хранимая процедура "Получить информацию о прокате по идентификатору":

CREATE PROCEDURE ПолучитьИнформациюОПрокатеПоId

@id\_проката INT

AS

BEGIN

SELECT \*

FROM Прокаты

WHERE id\_проката = @id\_проката

END

--Хранимая процедура "Обновить информацию о клиенте":

CREATE PROCEDURE ОбновитьИнформациюОКлиенте

@id\_клиента INT,

@Адрес VARCHAR(100),

@Телефон VARCHAR(15),

@Email VARCHAR(50)

AS

BEGIN

UPDATE Клиенты

SET Адрес = @Адрес, Телефон = @Телефон, Email = @Email

WHERE id\_клиента = @id\_клиента

END

--Хранимая процедура "Удалить прокат":

CREATE PROCEDURE УдалитьПрокат

@id\_проката INT

AS

BEGIN

DELETE FROM Прокаты

WHERE id\_проката = @id\_проката

END

CREATE TRIGGER ПередУдалениемАвтомобиля

ON Автомобили

INSTEAD OF DELETE

AS

BEGIN

-- Проверка наличия прокатов для удаляемого автомобиля

IF EXISTS(SELECT 1 FROM Прокаты WHERE id\_автомобиля IN (SELECT id\_автомобиля FROM DELETED))

BEGIN

RAISERROR ('Нельзя удалить автомобиль, имеющий активные прокаты', 16, 1)

END

ELSE

BEGIN

-- Удаление автомобиля

DELETE FROM Автомобили WHERE id\_автомобиля IN (SELECT id\_автомобиля FROM DELETED)

END

END

CREATE TRIGGER ПередВставкойАвтомобиля

ON Автомобили

INSTEAD OF INSERT

AS

BEGIN

-- Проверка наличия автомобиля с таким же регистрационным номером

IF EXISTS(SELECT 1 FROM Автомобили WHERE Регистрационный\_номер IN (SELECT Регистрационный\_номер FROM INSERTED))

BEGIN

RAISERROR ('Автомобиль с таким регистрационным номером уже существует', 16, 1)

END

ELSE

BEGIN

-- Вставка нового автомобиля

INSERT INTO Автомобили (Марка, Модель, Год\_выпуска, Цвет, Регистрационный\_номер)

SELECT Марка, Модель, Год\_выпуска, Цвет, Регистрационный\_номер FROM INSERTED

END

END

CREATE TRIGGER ПриВставкеАвтомобиля

ON Автомобили

AFTER INSERT

AS

BEGIN

PRINT 'Новый автомобиль добавлен в базу данных.'

END

CREATE FUNCTION ПолучитьАвтомобилиПоМарке(@марка VARCHAR(100))

RETURNS TABLE

AS

RETURN (

SELECT \*

FROM Автомобили

WHERE Марка = @марка

);

CREATE FUNCTION ПолучитьПрокатыПоКлиенту(@id\_клиента INT)

RETURNS TABLE

AS

RETURN (

SELECT \*

FROM Прокаты

WHERE id\_клиента = @id\_клиента

);

CREATE FUNCTION ПолучитьСотрудниковПоДолжности(@должность VARCHAR(50))

RETURNS TABLE

AS

RETURN (

SELECT \*

FROM Сотрудники

WHERE Должность = @должность

);

1. **Импорт данных пользователя в базу данных**

Импорт данных пользователя в базу данных — это процесс передачи данных, находящихся во внешних источниках (например, файлы CSV, Excel или другие форматы данных), в базу данных для сохранения и последующего использования. Это позволяет обновить или расширить содержимое базы данных с помощью внешних данных.

Вот основные шаги для импорта данных пользователя в базу данных:

1. Подготовка источника данных: Пользователь должен предоставить данные, которые необходимо импортировать в базу данных. Это могут быть файлы в форматах CSV, Excel, XML или другие форматы данных.

2. Подготовка базы данных: Проверьте структуру базы данных, чтобы убедиться, что она соответствует ожидаемым данным. Если необходимо, создайте таблицы или обновите существующие структуры данных, чтобы принять импортируемые данные.

3. Использование инструментов импорта: В большинстве СУБД существуют инструменты и функции для импорта данных. Это может быть встроенная функция СУБД или сторонние инструменты, которые позволяют импортировать данные из различных источников.

4. Настройка параметров импорта: в некоторых случаях могут быть доступны параметры, которые позволяют настроить процесс импорта данных. Например, можно указать разделитель полей в файле CSV или выбрать определенные столбцы для импорта из файла Excel.

5. Выполнение импорта: запустите процесс импорта, используя выбранные инструменты или команды. Данные из внешнего источника будут обработаны и загружены в базу данных с учетом заданных параметров.