CREATE DATABASE Ломбард;

GO

USE Ломбард;

GO

-- Создание таблицы "Категории"

CREATE TABLE Категории (

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

Название VARCHAR(100) NOT NULL,

Описание VARCHAR(200)

);

GO

-- Создание таблицы "Предметы"

CREATE TABLE Предметы (

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

Наименование VARCHAR(100) NOT NULL,

Описание VARCHAR(200),

Стоимость DECIMAL(10, 2) NOT NULL,

Дата\_приема DATE NOT NULL,

id\_категории INT,

FOREIGN KEY (id\_категории) REFERENCES Категории(id\_категории)

);

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,

Адрес VARCHAR(200) NOT NULL,

Телефон VARCHAR(20) 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 "Займы"

CREATE TABLE Займы (

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

id\_клиента INT,

id\_предмета INT,

id\_сотрудника INT,

Сумма DECIMAL(10, 2) NOT NULL,

Дата\_выдачи DATE NOT NULL,

Дата\_возврата DATE,

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

FOREIGN KEY (id\_предмета) REFERENCES Предметы(id\_предмета),

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

);

GO

INSERT INTO Категории (Название, Описание) VALUES

('Электроника', 'Техника и гаджеты'),

('Ювелирные изделия', 'Украшения и драгоценности'),

('Антиквариат', 'Старинные предметы и коллекционные вещи');

GO

INSERT INTO Предметы (Наименование, Описание, Стоимость, Дата\_приема, id\_категории) VALUES

('Смартфон', 'Модель XYZ, черный', 500, '2023-05-01', 1),

('Золотое кольцо', 'С бриллиантом', 2000, '2023-04-15', 2),

('Старинные часы', 'Ручная работа, 19 век', 1500, '2023-04-20', 3);

GO

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

('Иванов', 'Алексей', 'ул. Центральная 10', '1234567890', 'ivanov@example.com'),

('Петров', 'Дмитрий', 'ул. Пушкина 5', '0987654321', 'petrov@example.com'),

('Сидорова', 'Елена', 'ул. Гагарина 7', '9876543210', 'sidorova@example.com');

GO

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

('Смирнов', 'Иван', 'Менеджер', 2500),

('Кузнецов', 'Александр', 'Эксперт', 3000),

('Волкова', 'Мария', 'Кассир', 2000);

GO

INSERT INTO Займы (id\_клиента, id\_предмета, id\_сотрудника, Сумма, Дата\_выдачи, Дата\_возврата) VALUES

(1, 1, 2, 300, '2023-05-02', '2023-06-02'),

(2, 2, 1, 1500, '2023-04-16', '2023-05-16'),

(3, 3, 3, 1000, '2023-04-21', '2023-05-21');

GO

CREATE VIEW Информация\_о\_займах AS

SELECT Займы.id\_займа, Клиенты.Фамилия, Клиенты.Имя, Предметы.Наименование, Займы.Сумма, Займы.Дата\_выдачи, Займы.Дата\_возврата

FROM Займы

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

JOIN Предметы ON Займы.id\_предмета = Предметы.id\_предмета;

CREATE VIEW Список\_клиентов\_и\_займов AS

SELECT Клиенты.Фамилия, Клиенты.Имя, COUNT(Займы.id\_займа) AS Количество\_займов

FROM Клиенты

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

GROUP BY Клиенты.Фамилия, Клиенты.Имя;

CREATE VIEW Предметы\_электроника AS

SELECT Предметы.Наименование, Предметы.Стоимость, Предметы.Дата\_приема, Категории.Название AS Категория

FROM Предметы

JOIN Категории ON Предметы.id\_категории = Категории.id\_категории

WHERE Категории.Название = 'Электроника';

CREATE PROCEDURE Добавить\_предмет

@Наименование VARCHAR(100),

@Описание VARCHAR(200),

@Стоимость DECIMAL(10, 2),

@Дата\_приема DATE,

@id\_категории INT

AS

BEGIN

INSERT INTO Предметы (Наименование, Описание, Стоимость, Дата\_приема, id\_категории)

VALUES (@Наименование, @Описание, @Стоимость, @Дата\_приема, @id\_категории);

END;

CREATE PROCEDURE Выдать\_займ

@id\_клиента INT,

@id\_предмета INT,

@id\_сотрудника INT,

@Сумма DECIMAL(10, 2),

@Дата\_выдачи DATE,

@Дата\_возврата DATE

AS

BEGIN

INSERT INTO Займы (id\_клиента, id\_предмета, id\_сотрудника, Сумма, Дата\_выдачи, Дата\_возврата)

VALUES (@id\_клиента, @id\_предмета, @id\_сотрудника, @Сумма, @Дата\_выдачи, @Дата\_возврата);

END;

CREATE PROCEDURE Поиск\_клиента\_по\_Фамилии

@Фамилия VARCHAR(100)

AS

BEGIN

SELECT \*

FROM Клиенты

WHERE Фамилия = @Фамилия;

END;

CREATE TRIGGER Проверка\_суммы\_займа

ON Займы

AFTER INSERT

AS

BEGIN

DECLARE @максимальная\_сумма DECIMAL(10, 2);

SET @максимальная\_сумма = 10000;

IF EXISTS (SELECT 1 FROM inserted WHERE Сумма > @максимальная\_сумма)

BEGIN

RAISERROR ('Сумма займа превышает максимально допустимую.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

CREATE TRIGGER Удаление\_связанных\_займов\_при\_удалении\_клиента

ON Клиенты

AFTER DELETE

AS

BEGIN

DELETE FROM Займы

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

END;

CREATE TRIGGER Автоматическое\_Обновление\_Стоимости\_Предмета

ON Предметы

AFTER UPDATE

AS

BEGIN

IF UPDATE(Стоимость)

BEGIN

UPDATE Займы

SET Сумма = i.Стоимость

FROM Займы z

INNER JOIN inserted i ON z.id\_предмета = i.id\_предмета;

END

END;

CREATE FUNCTION Получить\_общую\_стоимость\_займов\_клиента

(

@id\_клиента INT

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

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

SELECT @общая\_стоимость = SUM(Сумма)

FROM Займы

WHERE id\_клиента = @id\_клиента;

RETURN @общая\_стоимость;

END;

CREATE FUNCTION Получить\_среднюю\_зарплату\_сотрудников

(

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @средняя\_зарплата DECIMAL(10, 2);

SELECT @средняя\_зарплата = AVG(Зарплата)

FROM Сотрудники;

RETURN @средняя\_зарплата;

END;

CREATE FUNCTION Получить\_количество\_предметов\_категории

(

@id\_категории INT

)

RETURNS INT

AS

BEGIN

DECLARE @количество\_предметов INT;

SELECT @количество\_предметов = COUNT(\*)

FROM Предметы

WHERE id\_категории = @id\_категории;

RETURN @количество\_предметов;

END;

1. **Какие современные инструментальные средства проектирования схемы базы данных используют?**

Существует несколько современных инструментальных средств для проектирования схемы базы данных. Вот некоторые из них:

1. ER-диаграммы: ER-диаграммы (диаграммы сущность-связь) являются одним из наиболее распространенных инструментов для моделирования баз данных. Они позволяют визуально представить сущности (таблицы), их атрибуты и связи между ними.

2. UML-диаграммы: Unified Modeling Language (UML) - это стандартный язык моделирования, который также может быть использован для проектирования баз данных. UML-диаграммы классов и диаграммы объектов могут быть применены для описания сущностей и их связей в базе данных.

3. CASE-средства: CASE (Computer-Aided Software Engineering) - это программные средства, которые предоставляют набор инструментов для анализа, проектирования и разработки систем, включая базы данных. Они обычно предлагают функциональности ER-моделирования, автоматической генерации кода, документирования и других задач, связанных с проектированием баз данных.

4. Программное обеспечение для моделирования данных: Существуют специализированные программные инструменты для моделирования данных, такие как MySQL Workbench, Microsoft Visio, Oracle SQL Developer Data Modeler и другие. Они предоставляют возможности для создания ER-диаграмм, генерации сценариев создания таблиц, управления связями и других аспектов проектирования баз данных.

5. Онлайн-сервисы: В последнее время стали популярны онлайн-сервисы, которые позволяют создавать и моделировать базы данных через веб-интерфейс. Примеры таких сервисов включают dbdiagram.io, draw.io и Lucidchart.