Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчет по лабораторной работе №2

По теме “ Проектирование и создание БД”

|  |  |
| --- | --- |
|  | Выполнил студент группы 053504  Белько Владислав Игоревич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Проверил: Гриценко Н.Ю. |

Минск 2023

## СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc148036813)

[2 DFD, ERD И IDEF1X диаграммы 4](#_Toc148036814)

[3 Создание базы данных 6](#_Toc148036815)

[Приложение А](#_Toc148036816) [(обязательное)](#_Toc148036817) [DFD диаграмма 11](#_Toc148036818)

[Приложение Б](#_Toc148036819) [(обязательное)](#_Toc148036820) [Концептуальная ERD диаграмма 12](#_Toc148036821)

[Приложение В](#_Toc148036822) [(обязательное)](#_Toc148036823) [Физическая ERD диаграмма 13](#_Toc148036824)

[Приложение Г](#_Toc148036825) [(обязательное)](#_Toc148036826) [IDEF1X диаграмма 14](#_Toc148036827)

## 1 ЦЕЛЬ РАБОТЫ

Целью данной лабораторной работы является выполнение задач, связанных с проектированием базы данных для проекта. Этот процесс включает в себя следующие этапы:

1. Разработать структуру базы данных.
2. Создание логической и физической модели базы данных, используя DFD, ERD и IDEF1.X диаграммы для визуализации.
3. Создание DDL скриптов для создания базы данных на основе разработанных моделей.
4. Создание базы данных с использованием ранее созданных DDL скриптов.

Эти действия позволят нам эффективно разработать базу данных, которая будет использоваться в рамках программного продукта.

## 2 DFD, ERD И IDEF1X ДИАГРАММЫ

Диаграммы DFD, ERD и IDEF1X являются мощными инструментами в области проектирования баз данных и моделирования информационных систем. Каждая из них имеет свои особенности и предназначение

**Диаграммы DFD.**

DFD используются для моделирования потоков данных и процессов в информационных системах. Они позволяют визуализировать, как данные перемещаются через систему и какие процессы их обрабатывают. Часто используются на начальных этапах проектирования для выявления ключевых процессов и потоков данных.

DFD включают в себя процессы, входные и выходные данные, потоки данных и хранилища данных.

**Диаграммы ERD.**

ERD представляет собой диаграмму, используемую для моделирования сущностей (объектов), их атрибутов и отношений между сущностями в базе данных. Она применяется для проектирования баз данных, описания структуры данных и их взаимосвязей.

**Диаграммы IDEF1X.**

IDEF1X – это нотация, разработанная для моделирования информационных систем, особенно для проектирования баз данных. Она уделяет особое внимание определению сущностей, атрибутов и отношений. Часто используется в разработке баз данных, чтобы обеспечить четкое определение сущностей и их атрибутов.

**Сравнение диаграмм.**

Сферы применения:

1. DFD ориентирован на анализ потоков данных и бизнес-процессов.
2. ERD используется для моделирования структуры данных и отношений в базе данных.
3. IDEF1X предоставляет формализованный способ определения структуры данных в базе.

Уровень детализации:

1. DFD фокусируется на высокоуровневом описании процессов и данных.
2. ERD предоставляет более детальное представление сущностей и их атрибутов.
3. IDEF1X предлагает более детальный и формализованный метод описания структуры данных.

Применение в разработке:

1. DFD полезен на ранних этапах анализа бизнес-процессов.
2. ERD необходим при проектировании баз данных.
3. IDEF1X используется для более глубокого и точного моделирования баз данных.

## 3 СОЗДАНИЕ БАЗЫ ДАННЫХ

Для разработки соответствующего программного продукта была выбрана SQL базы данных MS SQL Server. Это является стратегически важным шагом, который может оказать значительное влияние на производительность, масштабируемость и надежность.

SQL базы данных обеспечивают структурированный подход к хранению данных, позволяя определить схему данных, создавать таблицы, задавать типы данных и отношения между данными. Это особенно важно, когда у нас есть жесткие требования к структуре данных.

SQL базы данных также поддерживают транзакции, что гарантирует целостность данных. Это важно, например, при обработке финансовых транзакций.

Одним из преимуществ SQL является мощный язык запросов, который позволяет выполнять сложные операции над данными, фильтровать и агрегировать их. SQL базы данных также предоставляют средства для создания индексов и оптимизации запросов, что ускоряет доступ к данным. Это особенно важно для обеспечения высокой производительности.

Схема и миграции в SQL базах данных позволяют управлять изменениями в структуре данных, что делает разработку более управляемой.

MS SQL Server, в частности, обладает большим сообществом разработчиков и обширной экосистемой инструментов и расширений, что облегчает работу и предоставляет доступ к ресурсам и документации.

Кроме того, SQL базы данных масштабируются с помощью репликации и кластеризации, что позволяет адаптировать ваше приложение к росту нагрузки.

DDL скрипт для создания таблиц:

CREATE TABLE Roles

(

id INT PRIMARY KEY IDENTITY,

title VARCHAR(100) NOT NULL,

);

CREATE TABLE Permissions

(

id INT PRIMARY KEY IDENTITY,

title VARCHAR(100) NOT NULL

);

CREATE TABLE Roles\_permissions

(

id INT PRIMARY KEY IDENTITY,

role\_id INT NOT NULL,

permission\_id INT NOT NULL,

FOREIGN KEY role\_id REFERENCES Roles(id) ON DELETE CASCADE,

FOREIGN KEY permission\_id REFERENCES Permissions(id) ON DELETE CASCADE

);

CREATE TABLE Account

(

id INT PRIMARY KEY IDENTITY,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(15) NOT NULL,

role\_id INT NOT NULL,

FOREIGN KEY role\_id REFERENCES Roles(id) ON DELETE CASCADE

);

CREATE TABLE Users

(

id INT PRIMARY KEY IDENTITY,

nickname VARCHAR(25) NOT NULL UNIQUE,

date\_birth DATE NOT NULL,

account\_id INT NOT NULL,

FOREIGN KEY account\_id REFERENCES Account(id) ON DELETE CASCADE

);

CREATE TABLE Moderators

(

id INT PRIMARY KEY IDENTITY,

nickname VARCHAR(25) NOT NULL UNIQUE,

date\_birth DATE NOT NULL,

account\_id INT NOT NULL,

FOREIGN KEY account\_id REFERENCES Account(id) ON DELETE CASCADE

);

CREATE TABLE Logs

(

id INT PRIMARY KEY IDENTITY,

account\_id INT NOT NULL,

type VARCHAR(50) NOT NULL,

representation VARCHAR(255) NOT NULL,

date\_create: DATE NOT NULL

FOREIGN KEY account\_id REFERENCES Account(id) ON DELETE CASCADE

);

CREATE TABLE Categories

(

id INT PRIMARY KEY IDENTITY,

title VARCHAR(50) NOT NULL

);

CREATE TABLE News

(

id INT PRIMARY KEY IDENTITY,

category\_id INT NULL,

title VARCHAR(50) NOT NULL,

upload\_date DATE NOT NULL,

description VARCHAR(100) NULL,

content CLOB NOT NULL,

FOREIGN KEY category\_id REFERENCES Categories(id) ON DELETE CASCADE

);

CREATE TABLE News\_images

(

id INT PRIMARY KEY IDENTITY,

new\_id INT NOT NULL,

image\_id INT NOT NULL,

FOREIGN KEY new\_id REFERENCES News(id) ON DELETE CASCADE,

FOREIGN KEY image\_id REFERENCES Images(id) ON DELETE CASCADE

);

CREATE TABLE Articles

(

id INT PRIMARY KEY IDENTITY,

category\_id INT NULL,

title VARCHAR(50) NOT NULL,

upload\_date DATE NOT NULL,

description VARCHAR(100) NULL,

content CLOB NOT NULL,

FOREIGN KEY category\_id REFERENCES Categories(id) ON DELETE CASCADE

);

CREATE TABLE articles\_images

(

id INT PRIMARY KEY IDENTITY,

article\_id INT NOT NULL,

image\_id INT NOT NULL,

FOREIGN KEY article\_id REFERENCES Articles(id) ON DELETE CASCADE,

FOREIGN KEY image\_id REFERENCES Images(id) ON DELETE CASCADE

);

CREATE TABLE Videos

(

id INT PRIMARY KEY IDENTITY,

category\_id INT NULL,

title VARCHAR(50) NOT NULL,

upload\_date DATE NOT NULL,

description VARCHAR(100) NULL,

video\_file BLOB NOT NULL,

FOREIGN KEY category\_id REFERENCES Categories(id) ON DELETE CASCADE

);

CREATE TABLE News\_videos

(

id INT PRIMARY KEY IDENTITY,

new\_id INT NOT NULL,

video\_id INT NOT NULL,

FOREIGN KEY new\_id REFERENCES News(id) ON DELETE CASCADE,

FOREIGN KEY video\_id REFERENCES Videos(id) ON DELETE CASCADE

);

CREATE TABLE Articles\_videos

(

id INT PRIMARY KEY IDENTITY,

article\_id INT NOT NULL,

video\_id INT NOT NULL,

FOREIGN KEY article\_id REFERENCES Articles(id) ON DELETE CASCADE,

FOREIGN KEY video\_id REFERENCES Videos(id) ON DELETE CASCADE

);

CREATE TABLE Images

(

id INT PRIMARY KEY IDENTITY,

category\_id INT NULL,

title VARCHAR(50) NOT NULL,

upload\_date DATE NOT NULL,

description VARCHAR(100) NULL,

image\_file BLOB NOT NULL,

FOREIGN KEY category\_id REFERENCES Categories(id) ON DELETE CASCADE

);

CREATE TABLE Calculators

(

id INT PRIMARY KEY IDENTITY,

title VARCHAR(50) NOT NULL,

upload\_date DATE NOT NULL,

description VARCHAR(100) NULL,

content CLOB NOT NULL

);

CREATE TABLE Favourites

(

id INT PRIMARY KEY IDENTITY,

account\_id INT NOT NULL,

upload\_date DATE NOT NULL,

new\_id INT NULL,

article\_id INT NULL,

image\_id INT NULL,

video\_id INT NULL,

calculator\_id INT NULL,

FOREIGN KEY account\_id REFERENCES Account(id) ON DELETE CASCADE,

FOREIGN KEY new\_id REFERENCES News(id) ON DELETE CASCADE,

FOREIGN KEY article\_id REFERENCES Articles(id) ON DELETE CASCADE,

FOREIGN KEY image\_id REFERENCES Images(id) ON DELETE CASCADE,

FOREIGN KEY video\_id REFERENCES Video(id) ON DELETE CASCADE,

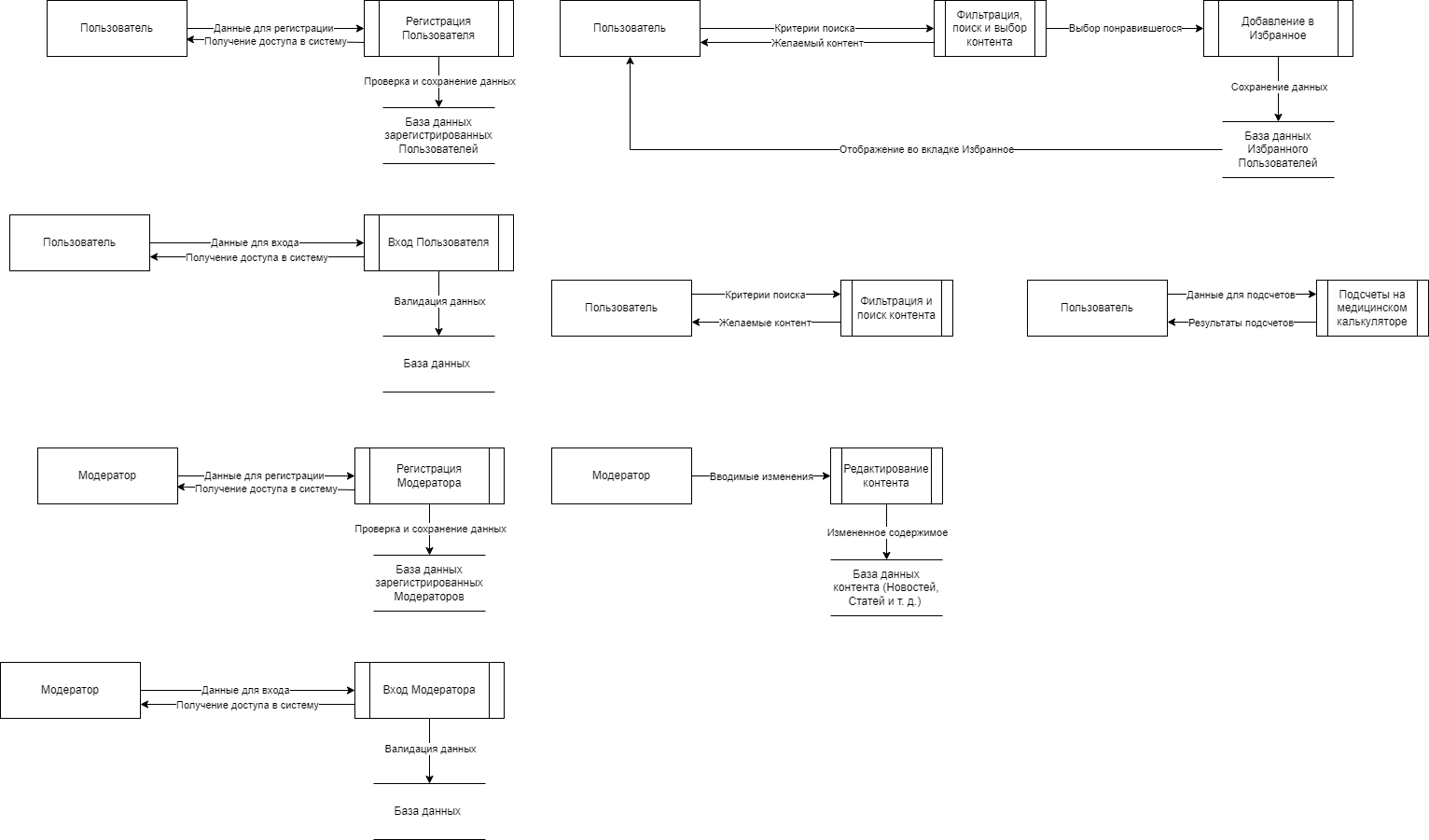
FOREIGN KEY calculator\_id REFERENCES Calculators(id) ON DELETE CASCADE

);

## ПРИЛОЖЕНИЕ А

## (обязательное)

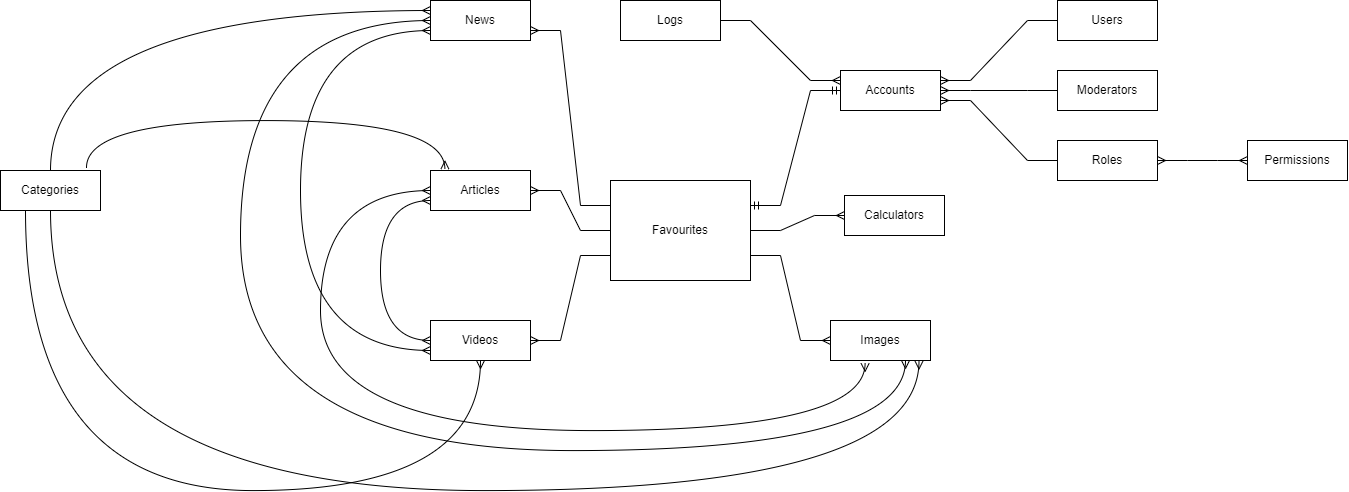
## DFD диаграмма



## ПРИЛОЖЕНИЕ Б

## (обязательное)

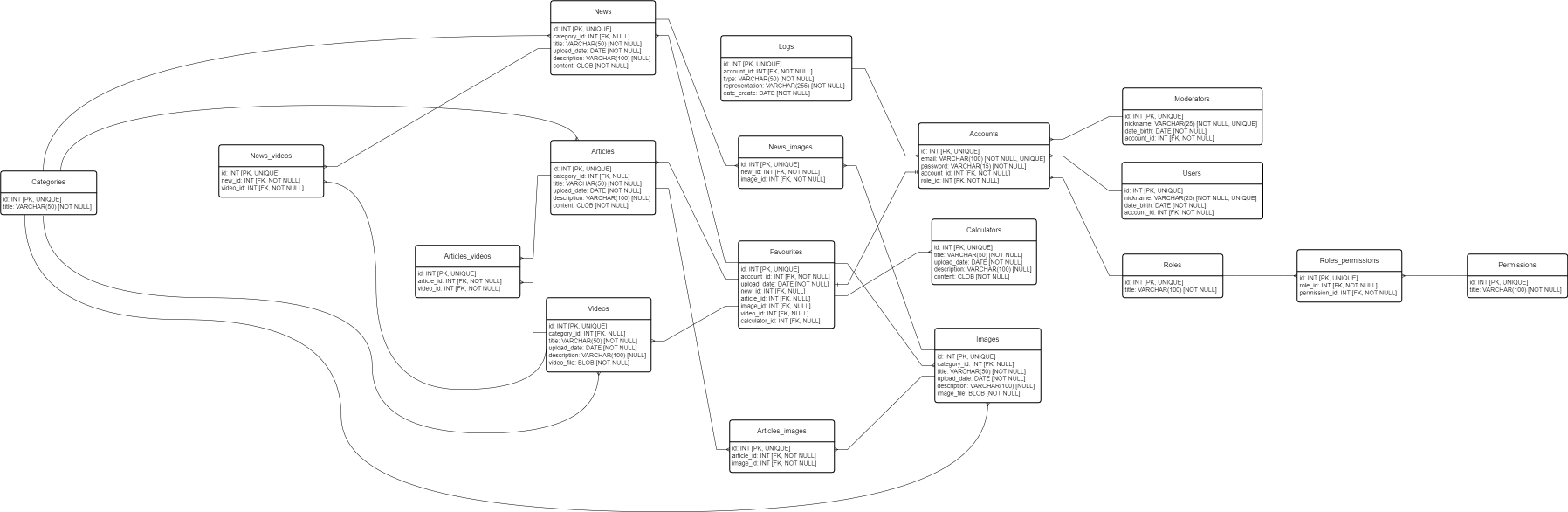
## Концептуальная ERD диаграмма



## ПРИЛОЖЕНИЕ В

## (обязательное)

## Физическая ERD диаграмма



## ПРИЛОЖЕНИЕ Г

## (обязательное)

## IDEF1X диаграмма

