**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**«Национальный исследовательский Томский политехнический университет»**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление «Информационные системы и технологии»

Отчет

по курсовому проекту №2

по дисциплине «Интернет-технологии»

**Модель данных**

Выполнил:

Студент группы 8И7А Толпикин Н.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

Доцент ОИТ Цапко С.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Томск 2021

**Цель работы**

Спроектировать модель БД для портала «Wiki-словарь».

**Описание портала «Вики-словарь»**

Словарь содержит слова и статьи, на одно слово может быть несколько статей. Статьям присвоены теги (один или несколько), определяющие их предметную область. На странице статьи должна быть информация о части речи и значении слова.

Функции портала:

С точки зрения неавторизованного пользователя:

* Пользователь может инициировать поиск по словарю, введя определенное слово с строку поиска
* Если статей на это слово несколько (омонимы), выводится список статей с тегами для перехода на страницу определенной статьи
* Если статья одна – открывается страница статьи

С точки зрения авторизованного пользователя, в дополнение к возможностям гостя:

* Пользователь может создавать статьи
* Пользователь может редактировать статьи
* Любое изменение (а также создание) статьи проходит модерацию и называется правкой
* Пользователь может просмотреть состояние правок для статей, которые он вносил (одобрены они или нет)
* Пользователь может комментировать правки

С точки зрения модератора, в дополнение к возможностям авторизованного пользователя:

* К каждой статье может быть привязано несколько модераторов
* Модератор может просматривать и комментировать правки для статей, привязанных к нему
* Модератор может одобрить правку, тогда текст оригинальной статьи изменится
* Все одобренные модератором правки сохраняются в истории изменения статьи

С точки зрения администратора, в дополнение к возможностям модератора:

* Администратор может добавлять, удалять редактировать и просматривать любые записи из любых сущностей (моделей) интернет-портала.

**Модели БД**

Для схем БД использовалось ПО Toad Data Modeller (TDM). В соответствии с вышеперечисленными функциями портала была реализована концептуальная схема БД (Рисунок 1).

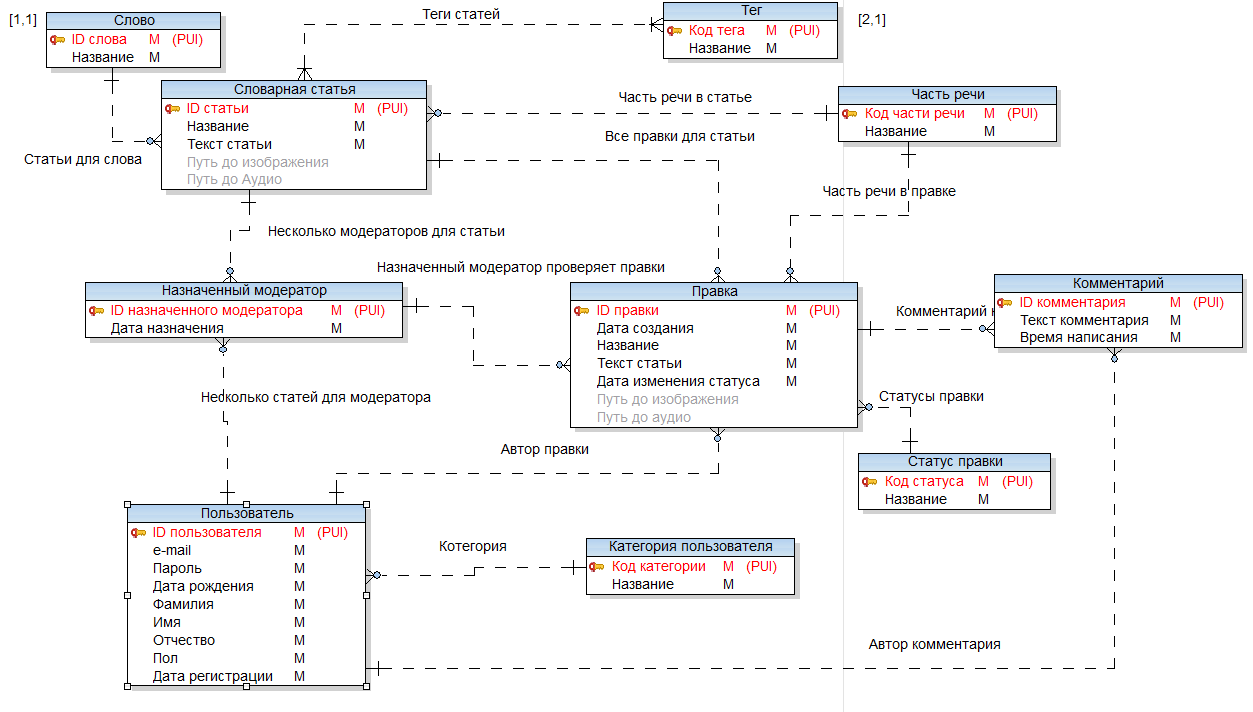


Рисунок 1 – Концептуальная схема БД в TDM

Далее концептуальная модель БД была конвертирована в логическую встроенными средствами TDM (Рисунок 2). На рисунке представлены таблицы, поля с типами данных и связи между таблицами в нотации «Воронья лапа». Красным цветом отмечены первичные ключи таблицы, зеленым – внешние ключи. Синим цветом отмечены внешние идентифицирующие ключи таблиц.

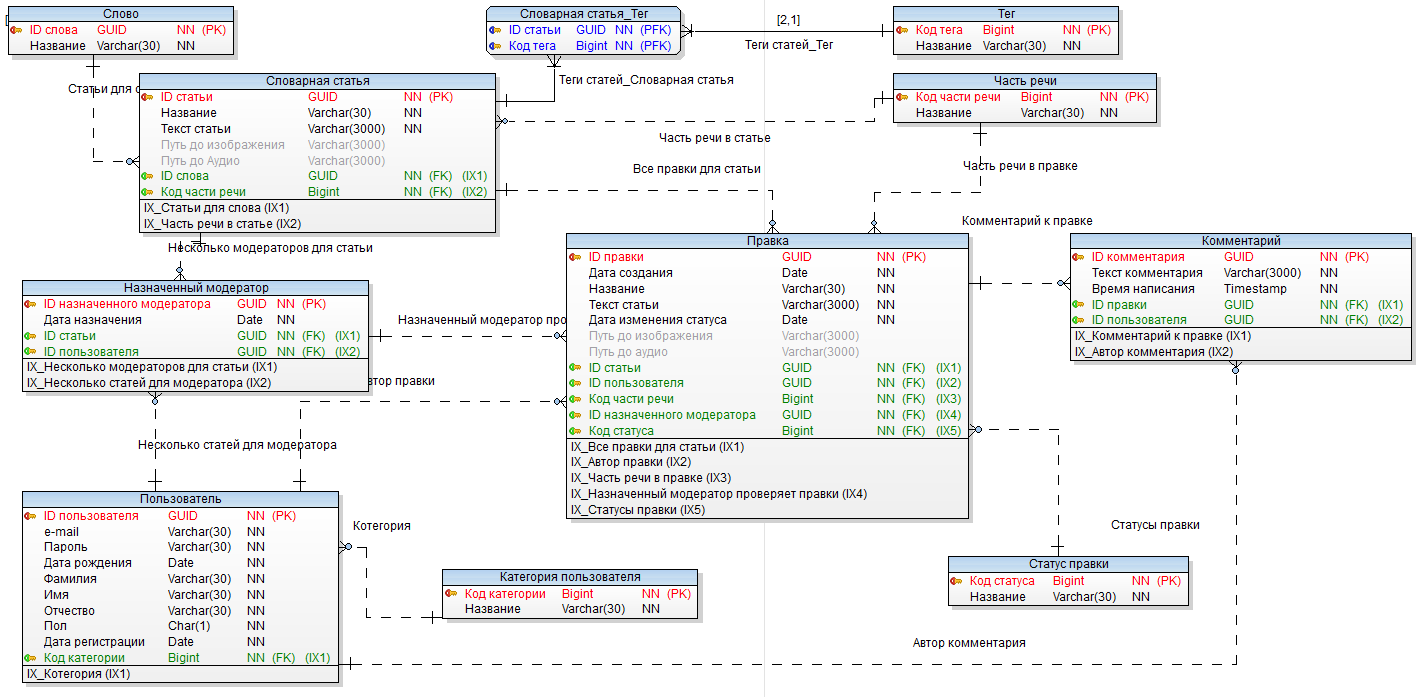


Рисунок 2 – Логическая модель БД в TDM

**Модели в MVC**

Для создания БД применяется паттерн Code-First для ORM. Это означает, что структура БД задается в коде приложения (классе контекста и моделях), а потом динамически создается в подключенной к веб-приложению БД средствами ORM. В этом проекте используется Entity Framework Core 2.1.4. Ссылки на модели представлены в Таблице 1.

Таблица 1 – Таблица ссылок на классы модели

|  |  |  |
| --- | --- | --- |
| **Название** | **Назначение** | **Ссылка** |
| ВикисловарьContext.cs | Класс контеста БД - задает структуру БД и связывает модели с их данными | [ВикисловарьContext.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%92%D0%B8%D0%BA%D0%B8%D1%81%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8CContext.cs) |
| КатегорияПользователя.cs | Задает модель для таблицы "Категория пользователя" | [КатегорияПользователя.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D1%8F%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.cs) |
| Комментарий.cs | Задает модель для таблицы "Комментарий" | [Комментарий.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B9.cs) |
| НазначенныйМодератор.cs | Задает модель для таблицы "Назначенный модератор" | [НазначенныйМодератор.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%9D%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9%D0%9C%D0%BE%D0%B4%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80.cs) |
| Пользователь.cs | Задает модель для таблицы "Пользователь" | [Пользователь.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C.cs) |
| Правка.cs | Задает модель для таблицы "Правка" | [Правка.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%9F%D1%80%D0%B0%D0%B2%D0%BA%D0%B0.cs) |
| СловарнаяСтатья.cs | Задает модель для таблицы "Словарная статья" | [СловарнаяСтатья.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D0%BD%D0%B0%D1%8F%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F.cs) |
| СловарнаяСтатьяТег.cs | Задает модель для таблицы "Словарная статья Тег" | [СловарнаяСтатьяТег.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D0%BD%D0%B0%D1%8F%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F%D0%A2%D0%B5%D0%B3.cs) |
| Слово.cs | Задает модель для таблицы "Слово" | [Слово.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%A1%D0%BB%D0%BE%D0%B2%D0%BE.cs) |
| СтатусПравки.cs | Задает модель для таблицы "Статус правки" | [СтатусПравки.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%A1%D1%82%D0%B0%D1%82%D1%83%D1%81%D0%9F%D1%80%D0%B0%D0%B2%D0%BA%D0%B8.cs) |
| Тег.cs | Задает модель для таблицы "Тег" | [Тег.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%A2%D0%B5%D0%B3.cs) |
| ЧастьРечи.cs | Задает модель для таблицы "Часть речи" | [ЧастьРечи.cs](https://github.com/NikitaTolpikin/Wiki/blob/main/Wiki/Models/%D0%A7%D0%B0%D1%81%D1%82%D1%8C%D0%A0%D0%B5%D1%87%D0%B8.cs) |

Для внесения изменений в структуру таблицы использовался модуль миграций Entity Framework Core. Для создания миграции достаточно внести изменения в класс контекста и применить в командной строке (в папке проекта) следующие команды:

* dotnet ef migration add <название\_миграции>
* dotnet ef database update

После этого будет создан класс с данными миграции и таблица в БД с информацией о проведенных миграциях. В качестве СУБД используется MS SQL 2019 EXPRESS.