**ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

На сайте планируется разработка раздела “Статьи”. Публикации должны иметь заголовок, анонсовое и детальное описание, а также дополнительные атрибуты (количество и состав атрибутов не ограничен). Необходимо подготовить структуру базы данных для раздела, если:

● каждая публикация должна быть размещена как минимум в одном подразделе, но может быть и в нескольких

● автором статьи могут быть несколько человек, каждый из которых может написать множество статей

● есть возможность добавить к каждой публикации комментарий (любым посетителем)

● каждая статья может быть оценена любым посетителем

● все публикации помечены тематическим тегом (для удобства группировки)

**ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

Для проектирования БД нужно руководствоваться нормальным формам (1-ую, 2-ую, 3-ю).

Отношение находится в 1НФ, если сохраняемые данные на пересечении строк и столбцов представляют скалярное значение, а таблицы не должны содержать повторяющихся строк..

Действительно, если проанализировать все таблице, то можно сделать вывод, что каждое поле таблицы содержит только одно значение, а также за счёт введения первичных ключей исключается возможность дублирования строк.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от первичного ключа. Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Для проверки нахождения схемы базы данных в 2НФ нам нужно рассмотреть таблицы, имеющие составные первичные ключи. Под данную характеристику подходят таблицы: «Publication\_Mark\_User» и «Publication\_Subsection», «Publication\_Author», «Comment». Действительно, эти таблицы находятся в 1НФ, т.к. дублирование строк отсутствует за счёт наличия первичного ключа, состоящего из внешних ключей, многозначных атрибутов не присутствует, также все атрибуты зависят от первичного ключа целиком, а не от какой-либо его части.

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут не транзитивно зависит от первичного ключа, это означает, что все атрибуты таблицы зависят от первичного ключа, но не от других атрибутов.

Таблицы удовлетворяют 3НФ, так как не имеют атрибутов, не относящихся к первичным или потенциальным ключам.

Таблицы «Tag» и «Publication» разделены на две и связаны с помощью внешних ключей потому, что название тега никак не зависит от первичного ключа таблицы «Publication». Один тег может характеризовать несколько публикация. Следовательно таблицы находятся в 3НФ.

Таблицы «Mark» и «Publication\_Mark\_User» разделены на две и связаны с помощью внешних ключей потому, что оценка никак не зависит от первичного ключа таблицы «Publication\_Mark\_User». Одна оценка может характеризовать несколько публикаций оцененных пользователем. Из всего вышеизложенного можно сделать вывод, что вся схема находится в 3НФ.

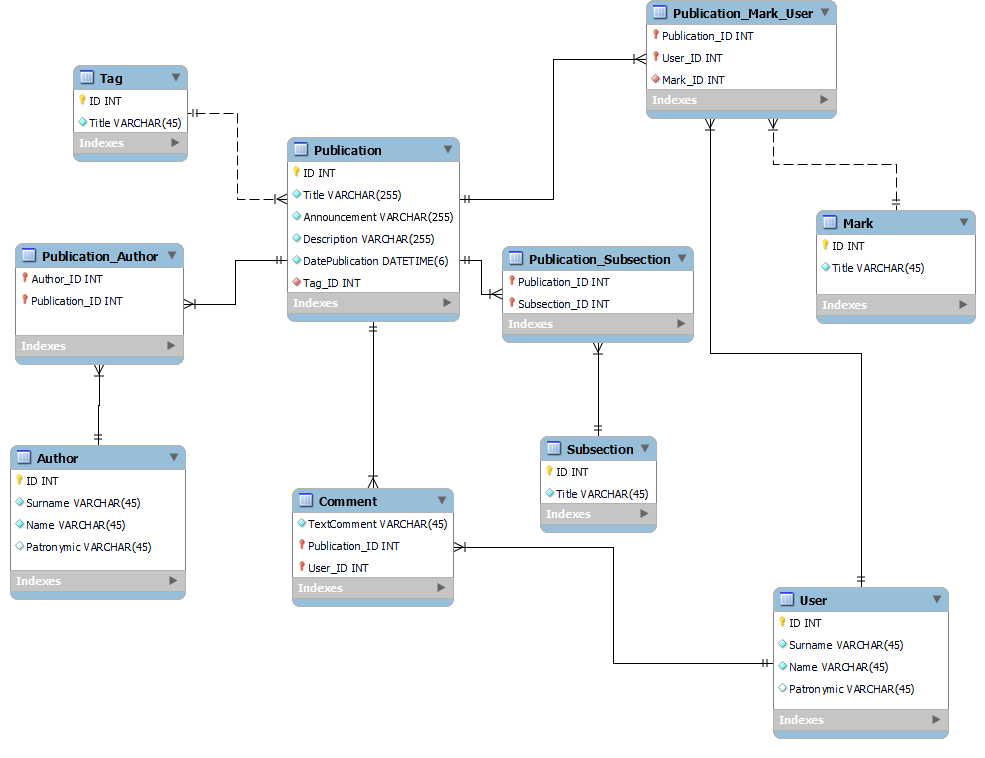
Необходимо решить проблему с зависимостью типа «многие ко многим» между таблицами «Publication» и «Author», «Publication» и «Subsection», «Publication» и «User». Например, между таблицами «Publication» и «Author» такая связь так как один автор может быть автором нескольких публикаций, а одной публикации – несколько авторов. Чтобы решить данный вопрос создаётся промежуточная таблица между ними «Publication\_Author», в которой будут два поля, являющиеся составным первичным ключом, включающий в себя внешние ключи на таблицы «Publication» и «Author». Аналогичным образом делается и для других таблиц, которые имеют связь «многие ко многим».

В проектируемой таблице рассматриваются следующие типы данных:

1. DATETIME — тип данных для хранения даты и времени. 1 Он включает день, месяц, год, а также часы, минуты и секунды;
2. INT – представляет целые числа от -2147483648 до 2147483647, занимает 4 байта;
3. VARCHAR() – представляет строку переменной длины. Длина хранимой строки также указывается в скобках, например, VARCHAR(10). Однако в отличие от CHAR хранимая строка будет занимать именно столько места, сколько необходимо;

Для реализации была выбрана СУБД MySQL.

Физическая схема базы данных представлена на рисунке 1.



**РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ**

После разработки физической схемы следующим шагом является создание базы данных. Ниже представлен скрипт создания таблицы «Publication», остальные созданы похожим образом.

CREATE TABLE IF NOT EXISTS mydb.Publication (

ID INT NOT NULL AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL,

Announcement VARCHAR(255) NOT NULL,

Description VARCHAR(255) NOT NULL,

DatePublication DATETIME(6) NOT NULL,

Tag\_ID INT NOT NULL,

PRIMARY KEY (ID),

INDEX `fk\_Publication\_Tag1\_idx (Tag\_ID ASC) VISIBLE,

CONSTRAINT fk\_Publication\_Tag1

FOREIGN KEY (Tag\_ID)

REFERENCES mydb.Tag (ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

Ниже представлен скрипт заполнения таблиц данными «Сomment», «Publication\_subsection» остальные созданы похожим образом.

INSERT INTO mydb.comment ('TextComment', 'Publication\_ID', 'User\_ID') VALUES ('Хорошая статья', '2', '1');

INSERT INTO mydb.publication\_subsection ('Publication\_ID', 'Subsection\_ID') VALUES ('2', '1');

**Задание:**

Написать SQL-запрос на получение всех оценок и комментариев к публикациям заданного автора.

select Publication.Title, Author.Surname, Author.Name, Author.Patronymic,

Comment.TextComment, Publication\_Mark\_User.Mark\_ID, User.Surname, User.Name, User.Patronymic

from Publication\_Author

join Publication on Publication\_ID = Publication.ID

join Author on Author\_ID = Author.ID

join Comment on Comment.Publication\_Id = Publication.ID

join Publication\_Mark\_User on Publication\_Mark\_User.Publication\_ID = Publication.ID

join User on Publication\_Mark\_User.User\_ID = User.id

where Author.Surname = 'Цветкова';

