**Создание форума**

**Задача 1**

Создать форум.

В информации о посте указать имя пользователя, дату сообщения, дату регистрации на форуме, дату последнего посещения форума, аватар.

Примерная БД форума с минимальным набором функций может быть как в примере.

**Задача 2\***

При входе с ролью Администратор иметь возможность модерировать форум

**Пример БД Forum**

**Таблица authors,** содержит различные данные о зарегистрированных посетителях форума:

ник (name); пароль (passw); e-mail (email); Web-адрес сайта посетителя (url); сведения посетителе (about); строку, содержащую путь к файлу фотографии посетителя (photo); время добавления запроса (putdate); последнее время посещения форума (last\_time); статус посетителя (statususer) — является ли он модератором ('moderator'), администратором ('admin') или обычным посетителем ('user'). Кроме перечисленных полей в таблице имеется поле id, представляющее собой первичный ключ таблицы.

Поля, в которых хранятся короткие строки (имя пользователя, его пароль, e-mail и т. п.), имеют тип данных TINYTEXT, позволяющий вводить не более 256 символов, можно объявлять их также с типом WARCHAR

CREATE TABLE authors (

id INT(11) NOT NULL AUTO\_INCREMENT,

name TINYTEXT,

passw TINYTEXT,

email TINYTEXT,

url TEXT,

about TEXT,

photo TINYTEXT,

putdate DATETIME DEFAULT NULL,

last\_time DATETIME DEFAULT NULL,

themes INT(10) DEFAULT NULL,

statususer ENUM('user','moderator','admin') NOT NULL default 'user', PRIMARY KEY (id)

);

**Таблица sections,** содержит данные о разделах форума.

Для удобства на форуме может быть создано несколько различных разделов. К примеру, на форуме, посвященном языкам программирования, чтобы не смешивать темы, относящиеся к различным языкам, имеет смысл создать следующие разделы: С++, PHP, Java т. д.

В таблице sections присутствуют следующие поля: первичный ключ (id); название раздела (name); правила форума (rule); краткое описание форума (logo); порядковый номер (pos) и флаг (hide), принимающий значение 'hide', если форум скрытый, и 'show', если он общедоступен.

CREATE TABLE sections (

id INT(11) NOT NULL AUTO\_INCREMENT,

name TINYTEXT,

rule TEXT,

logo TINYTEXT,

pos INT(11) DEFAULT NULL,

hide ENUM(‘show’,’hide’) NOT NULL DEFAULT ‘show’, PRIMARY KEY (id)

);

Структура форума может быть следующей: имеется список разделов, переход по которому приводит посетителя к списку тем раздела. При переходе по теме посетитель приходит к обсуждению этой темы, состоящему из сообщений других посетителей.

**Таблица themes,** содержит темы форума .

CREATE TABLE themes (

id INT(11) NOT NULL AUTO\_INCREMENT,

name TINYTEXT,

id\_author INT(11) DEFAULT NULL,

hide ENUM(‘show’,’hide’) NOT NULL DEFAULT ‘show’, putdate DATETIME DEFAULT NULL, forum\_id INT(11) default NULL,

PRIMARY KEY (id)

);

В таблице themes присутствуют следующие поля:

первичный ключ (id); название темы (name); внешний ключ к таблице авторов (author\_id); флаг (hide), принимающий значение 'hide', если тема скрыта, и 'show', если она отображается (это поле необходимо для модерирования); время добавления темы (putdate); внешний ключ к таблице форумов (forum\_id), для того чтобы определить, к какому разделу форума относится данная тема.

**Таблица posts,** в которой хранятся сообщения.

CREATE TABLE posts (

id INT(11) NOT NULL AUTO\_INCREMENT,

name TINYTEXT,

url TEXT,

file TINYTEXT,

author\_id INT(11) DEFAULT NULL,

hide ENUM(‘show’,’hide’) NOT NULL DEFAULT ‘show’, putdate DATETIME DEFAULT NULL, parent\_post INT(11) DEFAULT NULL,

theme\_id INT(11) DEFAULT NULL,

PRIMARY KEY (id)

);

В таблице posts присутствуют следующие поля: первичный ключ (id); тело сообщения (name); необязательная ссылка на ресурс, которую автор сообщения может ввести при добавлении сообщения (url); путь к файлу, прикрепляемому к сообщению (file), внешний ключ к таблице авторов (author\_id); флаг (hide), принимающий значение 'hide', если сообщение скрытое, и 'show', если он отображается (это поле необходимо для модерирования); время добавления сообщения (putdate); сообщение, ответом на которое является данное сообщение (parent\_post): если это первое сообщение в теме, то поле равно 0; внешний ключ к теме (theme\_id), указывающий, к какой теме относится сообщение.