Обоснование создания базы данных

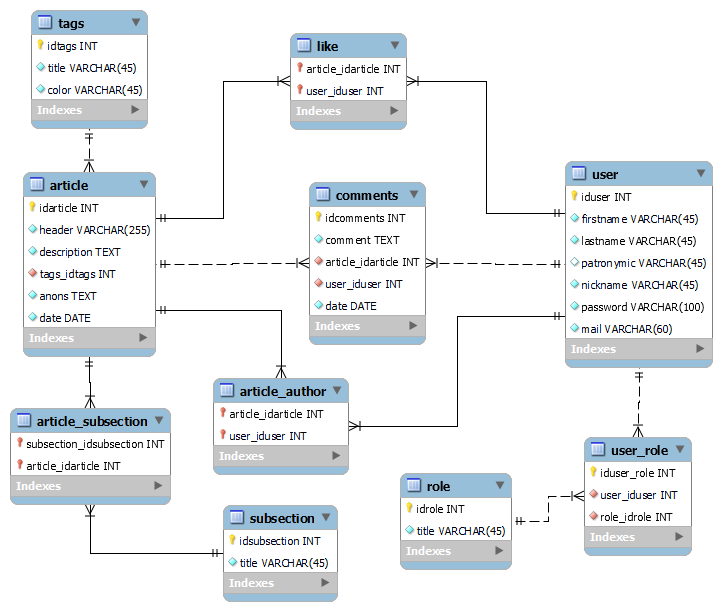


Таблица «article». В качестве уникально идентификатора было выбрано поле «idarticle». Эта таблица имеет внешний ключ от таблицы «tags», который был передан путём связи «один ко многим», то есть у статьи может быть лишь один тэг, но определённый тэг может быть у многих статей.

Таблица «article\_subsection». Эта таблица имеет первичный ключ, состоящий из внешних ключей от таблиц «article» и «subsection», которые были переданы путём связей «один ко многим», то есть статья может быть в нескольких подразделах, а в одном подразделе может быть несколько статей.

Таблица «subsection». Эта таблица имеет первичный ключ int. Поле «title» не походит в качестве уникального идентификатора, так как и название статьи может быть у нескольких статей. Поэтому был введён первичный ключ «idsubsection».

Таблица «tags». Поле «цвет» и «название» хоть и уникальные в таблице, но они имеет тип varchar, то есть сточный, что могло бы усложнить работу с подобным ключом по нескольким аспектам:

1) Строки занимают больше места в памяти, что может привести к замедленному выполнению запросов;

2) Строки имеют регистр, что в дальнейшем может повлечь за собой сложность поиска, сортировки, фильтрации данных, а также дупликаций одной и той же строки в другом регистре;

3) Строка может со временем измениться, что приведёт к изменению всех внешних ключей, ссылающихся на эту запись. Это может привести к проблемам с целостностью данных и ухудшению производительности базы данных.

Поэтому был введён первичный ключ «idtags».

Таблица «like». Эта таблица имеет первичный ключ, состоящий из внешних ключей от таблиц «article» и «user», которые были переданы путём связей «один ко многим», то есть статью могут оценить много пользователей, а один пользователь может оценить много статей.

Таблица «comments». Имеет два внешних ключа для связи пользователя и статьи. То есть каждый пользователь может отставить один комментарий под одной статьёй. Ни одно поле не походит в качестве уникального идентификатора, так как и дата и комментарий могут быть у нескольких записей и человек. Поэтому был введён первичный ключ «idcomments».

Таблица «article\_author». Эта таблица имеет первичный ключ, состоящий из внешних ключей от таблиц «article» и «user», которые были переданы путём связей «один ко многим», то есть статью могут писать много пользователей, а один пользователь может писать много статей.

Таблица «user». В качестве первичного ключа был также выбран id пользователя, так как остальные поля имеют тип varchar и не подходят в качестве первичного ключа по описанным выше причинам.

Таблица «user\_code». Эта таблица имеет первичный ключ, состоящий из внешних ключей от таблиц «role» и «user», которые были переданы путём связей «один ко многим», то есть у одного пользователя может быть несколько ролей, а одну роль могут иметь несколько пользователей.

Таблица «role». «Название» не было выбрано в качестве первичного ключа, так как является строкой.

Ограничения для базы данных

Нужно будет разработать процедуру, которая будет проверять роль пользователя. Если у пользователя роль «автор», то он сможет создавать статью, иначе – нет.

Запросы на заполнение базы данных данными

INSERT INTO user(firstname, lastname, patronymic, nickname, password, mail) VALUES

('Горбачёв', 'Геннадий', 'Рудольфович', 'zarevi4', 'vvediteParol', 'gorb1991@yakrasiviy.ru'),

('Букина', 'Тамара', 'Геннадьевна', 'bezMuzhikov', 'silaDobra', 'tambuk@yarkiy.ru'),

('Добрынин', 'Елесей', 'Силович', 'nickiti4', 'lubavaTop10iz10', 'stuchitktoto@neya.ru');

INSERT INTO `articles`.`user\_role` (`iduser\_role`, `user\_iduser`, `role\_idrole`) VALUES ('1', '1', '2');

INSERT INTO `articles`.`user\_role` (`iduser\_role`, `user\_iduser`, `role\_idrole`) VALUES ('2', '2', '1');

INSERT INTO `articles`.`user\_role` (`iduser\_role`, `user\_iduser`, `role\_idrole`) VALUES ('3', '3', '1');

INSERT INTO `articles`.`tags` (`idtags`, `title`, `color`) VALUES ('1', 'Наука', 'красный');

INSERT INTO `articles`.`tags` (`idtags`, `title`, `color`) VALUES ('2', 'Образование', 'зелёный');

INSERT INTO `articles`.`tags` (`idtags`, `title`, `color`) VALUES ('3', 'Развлечение', 'синий');

INSERT INTO `articles`.`article` (`idarticle`, `header`, `description`, `tags\_idtags`, `anons`, `date`) VALUES ('1', 'первая', 'не очень большой текст содержания статьи', '2', 'анонсик', '2024.10.20');

INSERT INTO `articles`.`article` (`idarticle`, `header`, `description`, `tags\_idtags`, `anons`, `date`) VALUES ('2', 'вторая', 'интрересный текст на много символов', '1', 'анонсик другой', '2024.10.20');

INSERT INTO `articles`.`article` (`idarticle`, `header`, `description`, `tags\_idtags`, `anons`, `date`) VALUES ('3', 'третья', 'весь важный и умный текст статьи', '3', 'ананас', '2024.10.20');

INSERT INTO `articles`.`article\_author` (`article\_idarticle`, `user\_iduser`) VALUES ('1', '1');

INSERT INTO `articles`.`article\_author` (`article\_idarticle`, `user\_iduser`) VALUES ('2', '1');

INSERT INTO `articles`.`article\_author` (`article\_idarticle`, `user\_iduser`) VALUES ('3', '1');

INSERT INTO `articles`.`like` (`article\_idarticle`, `user\_iduser`) VALUES ('1', '2');

INSERT INTO `articles`.`like` (`article\_idarticle`, `user\_iduser`) VALUES ('2', '3');

INSERT INTO `articles`.`like` (`article\_idarticle`, `user\_iduser`) VALUES ('3', '3');

INSERT INTO `articles`.`comments` (`idcomments`, `comment`, `article\_idarticle`, `user\_iduser`, `date`) VALUES ('1', 'Как познавательно!', '2', '3', '2024.10.23');

INSERT INTO `articles`.`comments` (`idcomments`, `comment`, `article\_idarticle`, `user\_iduser`, `date`) VALUES ('2', 'Это лучшее что я видела!!!!!', '1', '2', '2024.10.23');

INSERT INTO `articles`.`comments` (`idcomments`, `comment`, `article\_idarticle`, `user\_iduser`, `date`) VALUES ('3', 'Ну я бы написал умнее конечно, глупости', '3', '3', '2024.10.23');

INSERT INTO `articles`.`subsection` (`idsubsection`, `title`) VALUES ('1', 'Для умных');

INSERT INTO `articles`.`subsection` (`idsubsection`, `title`) VALUES ('2', 'Для здоровья');

INSERT INTO `articles`.`subsection` (`idsubsection`, `title`) VALUES ('3', 'Интересное');

**Запрос на выборку комментариев и оценок**

select art.header as 'название статьи', u.firstname as 'фамилия автора', u.lastname as 'имя автора',

com.comment as 'комментарий', ucom.firstname as 'кто написал комментарий',

(SELECT COUNT(\*) FROM articles.like AS l WHERE l.article\_idarticle = art.idarticle) as 'количество оценок'

from article as art

join article\_author on idarticle = article\_idarticle

join user as u on iduser=user\_iduser

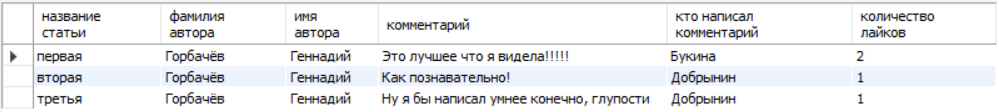
join comments as com on com.article\_idarticle = art.idarticle

join user as ucom on com.user\_iduser = ucom.iduser

join user\_role ur on u.iduser=ur.user\_iduser

join role r on r.idrole=ur.role\_idrole

where u.firstname = 'Горбачёв' and ur.role\_idrole = 2;



Количество оценок – количество лайков к записи. В реализованной базе данных на запись ставится один лайк.