## Пояснительная записка

**Стажёр – Котик Дмитрий Степанович**

**Тема – Новости**

Проектирование моделей необходимо для того, чтобы отразить элементы реального мира в информационной среде для дальнейшей работы с ними.

База данных будет являться реляционной. Эти данные представляют собой набор таблиц, которые связаны посредством первичных и внешних ключей.

Логическая модель данных предназначена для отражения объектов предметной области, указания их взаимосвязи, показывает ограничения, которые привносит предметная область. Данный тип модели является прототипом будущей базы данных. Она не привязывается к конкретной СУБД.

Физическая модель данных относится к самому низкому уровню и привязывается к конкретной СУБД. Все ограничения, представленные в логической модели, воплощаются средствами конкретной СУБД. Отношения, описанные в логической модели, становятся таблицами в физической модели, а атрибуты становятся столбцами таблицы. Результат всех этих моделей – сформированная база данных.

Объекты предметной области можно примерно разбить на несколько категорий:

* Новости:
  + Новость
  + Тег
  + Категория новости
  + Изображение новости
  + Комментарий к новости
  + Страница
* Пользователи и действия над объектами
  + Пользователь
  + Права пользователя
  + Логи действий
  + Контент
* Прочее
  + Рассылка
  + Сессия

Рассмотрим подробно каждую из них:

*Новости*

Центральным объектом здесь выступает объект новость. Данному объекту присущи свойства, которые характерны для большинства новостей. Данные для данного объекта хранятся в таблице **new**. Содержит следующие поля:

таблица **new:**

* id – суррогатный первичный ключ
* title – заголовок новости
* lead – лид новости
* create\_date – дата и время создания новости
* edit\_date – дата редактирования новости
* text – содержимое новости
* is\_published – флаг, указывающий на то, опубликована новость или нет
* category\_id – внешний ключ на таблицу категории. Связь – один-ко-многим
* tag\_id – внешний ключ на таблицу тегов. Связь – многие-ко-многим. Связь реализована через промежуточную таблицу **new\_tag**
* user\_id – внешний ключ на таблицу пользователей. Связь – один-ко-многим

Для хранения категорий используется таблица **category**. Одна новость может относиться только к одной категории. На неё ссылается таблица новостей.

таблица **category**.

* id – суррогатный первичный ключ
* title – название категории

Для хранения тегов используется таблица **tag**. Отличие тегов от категорий состоит в том, что одна новость может иметь множество тегов, как и один тег может быть присущ множеству новостей. На неё ссылается таблица новостей посредством связи многие-ко-многим.

таблица **tag**:

* id – суррогатный первичный ключ
* title – название тега

Как говорилось ранее, связь организована через дополнительную таблицу.

таблица **new\_tag:**

* id – суррогатный первичный ключ
* new\_id – внешний ключ на таблицу **new**
* tag\_id – внешний ключ на таблицу **tag**

Для хранения изображений новостям выделена отдельная таблица, т.к. для одной новости может храниться несколько изображений. Для изображений используется таблица **image**. В данной таблице хранятся пути к файлам изображений в файловой системе.

таблица **image**:

* id – суррогатный первичный ключ
* title – название изображения
* path – путь к изображению в файловой системе
* new\_id – внешний ключ на таблицу новостей **new**

Для хранения комментариев к новостям используется таблица **comment**.

таблица **comment:**

* id – суррогатный первичный ключ
* text – содержимое комментария
* create\_date – дата и время создания комментария
* edit\_date – дата и время редактирования комментария
* new\_id – внешний ключ на таблицу новостей **new**
* user\_id – внешний ключ на таблицу пользователей **user**

Для каждой новости должен быть выделен url, по которому будут доступны новости из категорий, тегов, и конкретная новость. Для этого создана таблица **page.** Она так же включает в себя различную служебную информацию, которая нужна тебя <head> страницы.

таблица **page:**

* id – суррогатный первичный ключ
* title – заголовок страницы
* meta\_charset – мета информация (кодировка страницы)
* meta\_description – мета информация (описание страницы)
* meta\_keywords – мета информация (ключевые слова страницы)
* title\_menu – название страницы в меню
* favicon\_path – путь к фавиконке в файловой системе
* is\_published – флаг, указывающий на то, опубликована страница или нет
* url – уникальный url страницы
* content\_id – внешний ключ на таблицу **content**

*Пользователи и действия над объектами*

В эту категория мы отнесем пользователей, их права на контент, сам контент и логи их действий.

Центральный объектом здесь выступает объект пользователь. Он включает в себя различную информацию, которая характерная для пользователя сайта. Все данные хранятся в таблице **user.**

таблица **user:**

* id – суррогатный первичный ключ
* last\_login – дата и время последнего входа
* is\_superuser – обладает ли пользователь правами администратора
* username – логин пользователя (никнейм)
* first\_name – имя пользователя
* last\_name – фамилия пользователя
* email – электронная почта пользователя
* is\_staff – является ли пользователь персоналом
* is\_active – активен пользователь или нет
* date\_jouned – дата и время создания пользователя

Каждый пользователь обладает определенным набором прав. На каждый объект существует 4 типа прав: чтение, изменения, создание, удаление. Все права перечислены в таблице **permission**. Так как у одного пользователя может быть множество прав, а у одного права может быть присуще множеству пользователей, то связь между таблицей прав **permission** и таблицей пользователей **user** является многие-ко-многим, и реализована с помощью промежуточной таблицы **user\_permission.**

таблица **permission**:

* id – суррогатный первичный ключ
* content\_id – внешний первичный ключ на таблицу **content**. Указывает, для какого типа сущности предоставлено право
* action – название права
* permission – флаг указывающий на то, предоставлено ли право на объект

таблица **user\_permission:**

* id – суррогатный первичный ключ
* user\_id – внешний ключ на таблицу пользователей **user**
* permission\_id – внешний ключ на таблицу прав **permission**

Для хранения истории действий пользователей используется таблица **log.**

таблица **log:**

* id – суррогатный первичный ключ
* content\_id – внешний ключ на таблицу объектов **content**
* user\_id – внешний ключ на таблицу пользователей **user**
* action – название операции
* action\_time – время совершения операции

Для хранения названий сущностей, на которые можно обладать правами и идентификации объектов в логах используется таблица **content.**

В идеале новостной сайт обладает множеством типов сущностей (классов объектов). Для полноценного функционирования этого типа сущности необходимо реализовывать механизм миграций, который позволит обновлять эту таблицу при добавлении новых классов. Но в связи с тем, что тема “Новости” на данном этапе не включает в себя реализацию таких сложных вещей, то ограничимся тем, что в таблицу **content** будет добавлен заранее определенный список классов, над которыми будут производиться действия (category, new, page и тд).

таблица **content:**

* id – суррогатный первичный ключ
* entity – сущность (или название класса)

*Прочее*

Данная категория включает в себя всего две таблицы. Вынесены они потому, что они не имеют внешних ключей.

Для нормальной работы клиент-серверного приложения нужно реализовать механизм сессий. Данные сессий будут храниться в таблице БД **session.**

таблица **session:**

* session\_key – естественный первичный ключ
* session\_data – данные сессии
* expire\_date – дата окончания действия сессии

Последней таблицей является **mailing**. Она предназначения для хранения email адресов пользователей, которые подписались на рассылку новостей.

таблица mailing:

* id – суррогатный первичный ключ
* email – адрес электронной почты пользователя

В данном разделе мы рассмотрели все таблицы модели базы данных.

Построение схемы велось в Visio.

Так как в Visio нет драйвера для проектирования БД для работы с PostgreSQL, то построение велось с использованием универсального драйвера ODBC. В связи с этим физическая модель будет немного отличаться от PostgreSQL.

Все первичные ключи помечены как **In**

Все ограничения уникальности помечены как **Un**

Первичные ключи – **PK**

Внешние ключи – **FKn**

Стрелка всегда указывает на родительскую сущность

Сущность, из которой выходит стрелка – дочерняя

Схема БД так же представлена отдельным файлом: **physical\_database\_schema**

