

УДК 004.652

**ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ ДЛЯ
ПРОГРАММНОГО КОМПЛЕКСА SSR**

Черкасов В.В., студент гр. ПИБ-211, IV курс

Научный руководитель: Верёвкин С.А., старший преподаватель
Кузбасский государственный технический университет имени Т.Ф.
Горбачева, г. Кемерово

Проектирование любой системы, так или иначе связанной с хранением данных, стоит начинать именно с определения того, как и где будет осуществляться хранение - и программный комплекс SSR не исключение.

Основные требования к СУБД

Чтобы выбрать СУБД, требуется составить список требований:

- Единицей хранения является ссылка на интернет-ресурс (публикацию на нём) и метаданные к ней, например:
 - Название публикации;
 - Ссылка на публикацию;
 - Дата публикации;
 - Автор публикации;
 - Набор тэгов, описывающих публикацию (со стороны ресурса-источника);
 - Набор тэгов, описывающих публикацию (созданные пользователем комплекса SSR);
 - Примечания/комментарии - пользовательское описание.

При этом разные интернет-ресурсы могут обладать различными метаданными, например: если это видео, то будет ещё поле “Длительность”. Добавлять в структуру БД каждое отдельное возможное поле для всех записей публикаций - пустая трата ресурсов, поэтому СУБД должна иметь возможность хранить данные с динамическим количеством полей;

- Хранение самих записей планируется организовать в виде иерархии, наподобие файловой системы: директории, а в них другие директории и файлы (записи о публикациях).

При этом директории тоже имеет свои метаданные.

Выбор вида СУБД

СУБД подразделяются на различные виды, в зависимости от того, в каком виде они хранят и предоставляют данные:

Реляционные

Реализация хранения в реляционной СУБД иерархических данных - сложна. Большую роль в усложнении играет язык запросов SQL, ведь для сложной структуры придётся составлять не менее сложные SQL-запросы.

Вдобавок использование в реляционных СУБД данных с динамическим количеством и типизацией полей ещё больше усложняет задачу.

Хоть многие реляционные СУБД и способны хранить “документы” - текстовые файлы как тип данных, но использование их не сильно упрощает задачу.

Объектно-ориентированные базы данных

Как и реляционные СУБД требуют жёсткой типизации, хоть и удобнее с точки зрения использования из кода.

Документо-ориентированные и контентные СУБД

Эти СУБД предназначены для хранения иерархически упорядоченных данных. Документо-ориентированные - хранение документов, контентные - хранение “всего”.

Эти СУБД отлично подходят под требования.

Остальные виды СУБД являются узкоспециализированными и не подходят под поставленную задачу.

Выбор СУБД

Среди них можно выделить следующих представителей:

MongoDB

как самую популярную документо-ориентированную;

YTsaurus

как российскую разработку;

ModeShape

как универсальное решение;

Своя СУБД поверх файловой системы

как низкоуровневое и наиболее гибкое решение.

Определение ограничений СУБД

Для проектирования БД требуется определить ограничения, которые ставят СУБД.

Ограничения на названия:

MongoDB

В названии может участвовать любой UTF-8 символ, кроме:

- * /
- * \
- * .
- * “
- * \$
- * \0
- * ”system.“ в начале названия
- * ”.system.“ в середине названия

YTsaurus

Информация по ограничениям на символы в названиях в документации не найдено.

ModeShape

С бэкендом в виде файловой системы, соответствует ограничениям файловой системы. Примем следующие ограничения:

- * /
- * \0

Примем за общие ограничения, предоставляемые MongoDB

Ограничения на вложенность/длину названий:

- Наибольшая длина полного имени у всех составляет в 1024 символа;
- Ограничения на длину одного имени - 255 байт или не более 63 UTF-8 символов.

Явных ограничений на количество файлов в директориях не замечено.

Примем за 3 000 000 файлов в директории и 60 000 для директорий в директориях на основе ограничений популярных файловых систем.

Во избежание сильной деградации производительности ограничим 1 000 000 файлами и 1 000 директорий.

Проектирование структуры БД



Рисунок 1. Структура БД

Готовая структура БД изображена на рисунке 1.

Стоит упомянуть, что здесь обозначение "[Пользователь]" обозначает множество директорий/файлов.

В корне структуры БД располагаются директории для данных самой системы (серверные данные), для глобальных переменных аддонов и данных пользователей.

Внутри пользовательской директории имеется разделение на: пользовательские данные аддонов, пользовательские переменные и непосредственно структура данных с вкладками.

Часть структуры данных, в которой хранятся вкладки требует дополнительного пояснения:

Системные директории

Системные директории и файлы в этой части иерархии отделяются от пользовательских точкой в начале названия, соответственно, пользователь не может ввести название для директории с точкой в начале.

В операционных системах семейства *nix файлы с точкой в начале определяются как скрытые, поскольку одна СУБД (а, возможно, в будущем больше) может использовать файловую систему как хранилище, то такое разделение на системные и пользовательские файлы вполне логично.

Директория

Разделение на пользовательский корень и ещё один уровень директорий позволит в будущем реализовать профили для пользователей.

Поддиректория

Пользователь может создавать поддиректории с учётом ограничений по названиям, о которых упоминалось ранее.

Созданные пользователем поддиректории имеют ту же структуру, что и верхняя директория.

Вкладки

Вкладки в директории представляют собой совокупность json-файлов, где один файл = одна вкладка.

Называются файлы вкладок согласно числовому ID, записанному в Base64 кодировке, где дополнительные 2 символа - '[' и ']', поскольку использовать '/' нельзя ни в какой из ранее упомянутых систем хранения данных.

Использование Base64 вместо Base10 значительно уменьшит потребление вычислительных ресурсов и памяти.

Позиция вкладок

Отдельная директория выделяется под распределение позиций вкладок, их сортировки.

Это требуется, поскольку пользователь должен иметь возможность сортировать вкладки в нужном ему порядке, в том числе в ручном. Также отделение сортировки от самих вкладок позволит функционалу публикации директории как включать пользовательскую сортировку в неё, так и предоставить получающему возможность выбрать порядок самому.

Метаданные директории

Здесь могут содержаться как системные данные, так и пользовательские. Разделение по директориям не требуется, поскольку в этой части иерархии для этого применяется точка в начале названия.

Полученная структура БД будет использована для разработки остальной системы.

Список литературы:

Виды баз данных. Большой обзор типов СУБД / Habr. — 21.08.2023. — URL: <https://habr.com/ru/companies/amvera/articles/754702/> (дата обр. 25.03.2025).

MongoDB Documentation. — URL: <https://www.mongodb.com/docs/> (дата обр. 25.03.2025).

Документация YTsaurus. — URL: <https://ytsaurus.tech/docs/ru/> (дата обр. 25.03.2025).

ModeShape 5 Documentation. — URL: <https://docs.jboss.org/author/display/MODE50/> (дата обр. 25.03.2025).