

СТРУКТУРНАЯ МОДЕЛЬ ХРАНЕНИЯ ПОЛИМОРФНЫХ ДАННЫХ В ФОРМАТЕ JSON КАК ОСНОВА ПРОЕКТИРОВАНИЯ МЕХАНИЗМА ОРГАНИЗАЦИИ ПРОСТРАНСТВЕННО- ВРЕМЕННЫХ ДАННЫХ

М.С. Кедрина, А.А. Волкова, А.В. Аргучинцев

Для организации полиморфного механизма хранения и представления данных целесообразно определить архитектуру ядра платформы не только в соответствии с указанным принципиальным разделением, но и с учетом требований:

- 1) динамического проектирования интерфейса для режима отображения веб-сайта по логике одностраничного приложения (SPA) [23, 24];
- 2) адаптации и кастомизации интерфейсов;
- 3) проектирования механизмов хранения произвольных внутренних баз данных;
- 4) средств аналитического описания пользовательских скриптов обработки данных;
- 5) регулируемой модели прав доступа к научным данным.

Проектирование интерфейсов научно-исследовательской платформы «BaikalIntelli» может обладать индивидуальными характеристиками программных интерфейсов и способов представления и организации пространственно-временных данных, что может определяться внутренними предпочтениями и требованиями научно-исследовательского направления. Поэтому в проектную модель серверной логики универсального метамеханизма хранения и систематизации полиморфных предлагается заложить паттерн модели приложения по архитектуре MVSC (Model-View-Service-Controller), реализующий логику динамической организации структур хранения данных, а также кастомизацию интерфейсов вывода и обработки данных. Такой концептуальный подход формирования динамических интерфейса позволит управлять:

- произвольным конструированием блоков рабочей области исследователя в рамках научной секции данных;
- созданием произвольных табличных представлений серий данных;
- организацией элементов интерфейса фильтра полей данных для возможности интерактивного формирования срезов и выборок данных, а также агрегирования информации в пространственно-временных базах данных в соответствии с предъявляемыми требованиями анализа данных;
- пользовательскими механизмами обработки данных.

Для реализации модели организации полиморфных данных предложена схема моделей хранения данных, состоящая из трех связанных таблиц, представленная на рис. 1.

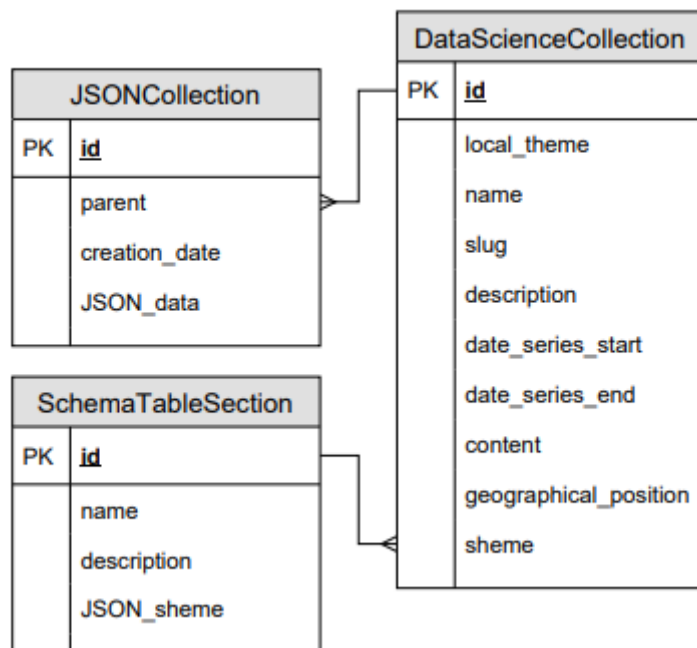


Рис. 1. ER-диаграмма структуры организации полиморфных данных

Модель DataScienceCollection предназначена для описания свойств произвольной пространственно-временной таблицы и представляет модель механизма коллекции полиморфных данных. Модель SchemaTableSection предназначена для формирования схемы хранения коллекции. Модель данных «SchemaTableSection» предназначена для универсального описания схемы хранения табличных данных с помощью формата JSON. В самом простом варианте структура JSON-кортеж для описания схемы состоит из массива вложенных словарей, содержащих ключи:

- name[I] – имя реквизита;
- type[I] – тип реквизита;
- column[I] – номер колонки.

Модель JSONCollection хранит экземпляры записей в соответствии для - таблица набор-записей исследовательских данных в рамках заданного шаблона схемы, описываемой в модели SchemaTableSection. Модель JSONCollection определяется реквизитами:

- table (ForeignKey DataScienceCollection) - связь с таблицей эксперимента;
- id - идентификатор строки;
- date - дата записи;
- JSON_attribute(TextField) - JSON-кортеж в соответствии со схемой атрибутов в таблице эксперимента.

Вид JSON-кортежа для хранения экземпляров записи, определяется массивом вложенных словарей, состоящих из ключей:

- name[I] – имя реквизита;

- value[I] – значение реквизита.

Пример спецификации описания схемы хранения табличных данных с помощью полей типа JSON приведен на рис. 2. Слева на рисунке показана спецификация организации реквизитов табличных данных, а справа пример хранения строки-кортежа в формате JSON.

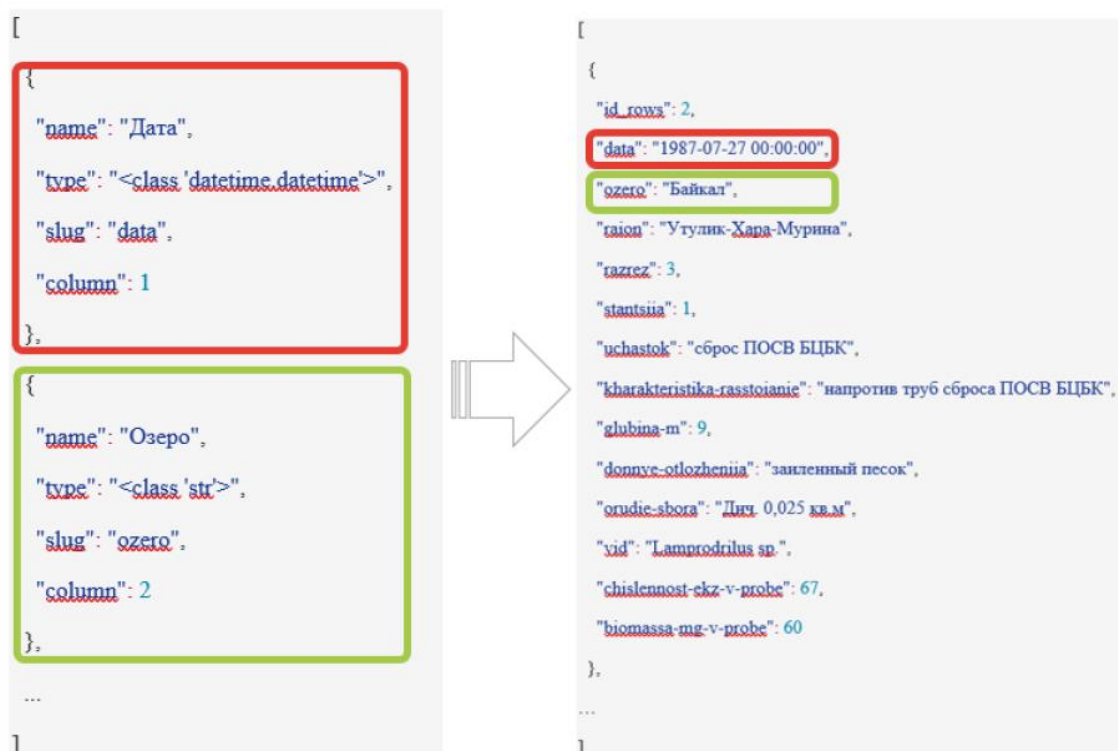


Рис. 2. Пример схемы хранения полиморфных данных

Таким образом, на уровне моделей «SchemaTableSection» и «JSONCollection» реализован механизм универсального хранения данных. Ключевой особенностью организации данных моделей является наличие особого типа полей и механизмов работы с форматом JSON, который позволяет иерархически описывать данные в виде комбинаций произвольных структур данных как:

- коллекций пар ключ/значение, которые на уровне языков программирования могут быть интерпретированы как структура или словарь;
- упорядоченных списков значений, которые на уровне языков программирования могут быть интерпретированы типами данных: «массив», «список» или «последовательность».