

УНИВЕРСАЛЬНАЯ МОДЕЛЬ ХРАНЕНИЯ ДАННЫХ КАК СРЕДСТВО КЛАССИФИКАЦИИ ПРИ РЕШЕНИИ ИССЛЕДОВАТЕЛЬСКИХ ЗАДАЧ

© 2016 Д.Е. Яблоков

Самарский национальный исследовательский университет имени академика С.П. Королёва

Статья поступила в редакцию 11.11.2016

В статье рассматривается концепция универсального хранилища данных, предназначенного для приложений, ориентированных на специалистов, занимающихся научными исследованиями. Приводятся определения и примеры использования механизмов типизации и категорирования как средств внутренней и внешней классификации, основанных на применении внутренних и внешних признаков соответственно. Предполагается, что структура хранилища не будет жёстко привязана к какой-либо предметной области, а будет востребована во многих проектах, связанных с накоплением, анализом и обработкой данных. Предлагаемый подход к организации системы хранения опирается на базовые принципы, концепции и технологии, используемые в процессе проектирования программного обеспечения, основанном на анализе предметной области и обнаружении критериев общности и изменчивости, являющихся гарантией хорошей техники абстрагирования – одного из основных инструментов для работы с данными.

Ключевые слова: универсальная модель данных, структура хранения, уровень абстракции, типизация данных, внутренняя классификация, категорирование данных, внешняя классификация.

ВВЕДЕНИЕ

Создание компьютерных систем для научных исследований – дело далеко не простое. По мере того как возрастает их сложность, процессы конструирования соответствующих программных продуктов становятся всё более трудоёмкими, причём затраты на разработку растут экспоненциально. Это относится и к процессу проектирования и реализации подсистем хранения, необходимых для функционирования приложений, используемых в научной работе. Исследователям, проводящим эксперимент, необходимо получать и обрабатывать достоверную и полную информацию, относящуюся к научным проблемам. Но на процесс подобных исследований негативно влияют обстоятельства, требующие изменения структур хранения данных об исследуемой предметной области или решаемой задаче. Среди них: необходимость тщательного подбора в соответствии с форматом хранения экспериментальных данных, проведение комплексных исследований с использованием разнотипной исходной информации и, как следствие, необходимость ее интерпретации или нормализации для использования в рамках эксперимента. Преодолеть вышеуказанные недостатки можно осуществив выход на качественно новый уровень разработки, создав универсальную модель [1] хранения и обработки разнородной информации. Такая модель хранения могла бы служить основой или определенным паттерном [2] для построения различных информационно-

вычислительных систем и формирования среды накопления формализованных данных для их дальнейшего применения в любых областях экспериментальных наук. Как и в других областях науки, в процессе создания систем такого рода, наиболее подходящей основой является дедуктивный метод. Он обеспечивает декомпозицию сложных понятий на более простые компоненты с математически и семантически обоснованным поведением. Чтобы пояснить, из чего состоят предлагаемые решения и методологии, основанные на фундаментальных понятиях программирования и анализа предметной области, необходимо дать краткий обзор некоторых категорий идей, которые соответствуют этим понятиям.

УНИВЕРСАЛЬНАЯ МОДЕЛЬ ХРАНЕНИЯ

Объекты любого типа и их отношения могут быть описаны в терминах некоторой упрощенной концептуальной модели (рис. 1) основными понятиями которой являются: существительное (Noun), прилагательное (Adjective), глагол (Verb), наречие (Adverb) и дескриптор (Descriptor) – лексическая единица служащая для описания основного смыслового значения и однозначно ставящаяся в соответствие группе ключевых понятий рассматриваемой предметной области.

В случае когда подобные понятия определены, несложно перейти к более конкретному описанию распространяя семантику данной концептуальной модели на объекты, их атрибуты (рис. 2), отношения между объектами и атрибуты отношений (рис. 3), применяя объектно-реляционный подход [3]. Суть этого подхода состоит в том, что при сохранении реляционного ядра

*Яблоков Денис Евгеньевич, ведущий инженер-программист Межвузовского научно-исследовательского центра по теоретическому материаловедению.
E-mail: dyablokov@gmail.com*

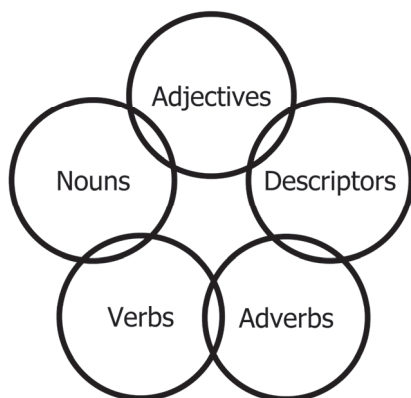


Рис. 1. Диаграмма универсальной модели данных

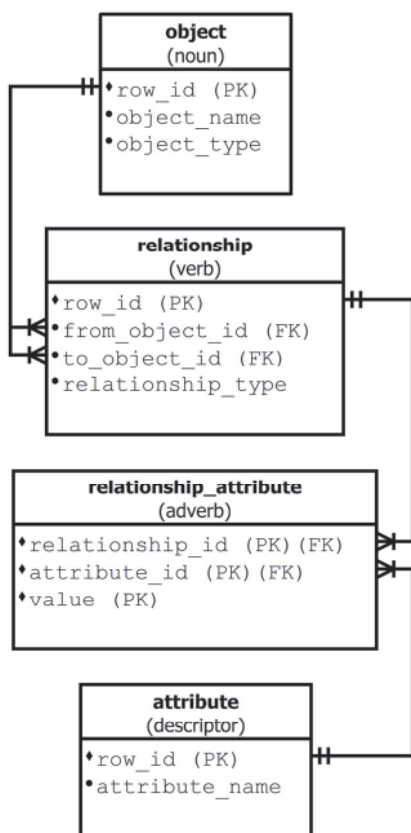


Рис. 3. Объектно-реляционный подход. Атрибуты отношений объектов.

системы хранения она наращивается более или менее удачными объектными надстройками. В качестве таких надстроек могут выступать, и расширяемая пользователем система типов, и средства описания иерархически взаимосвязанных данных, такие как наследование или композиция.

Этот пример является очень упрощенным, но, в тоже время, он может быть достаточно общим, чтобы поддерживать большинство приложений и обеспечивать добавление данных любого типа без указания конкретных имен таблиц или полей, которые необходимы при использовании реляционной модели данных в чистом виде [4]. В дополнение к этому важно отметить, что при использовании универсальной модели становится

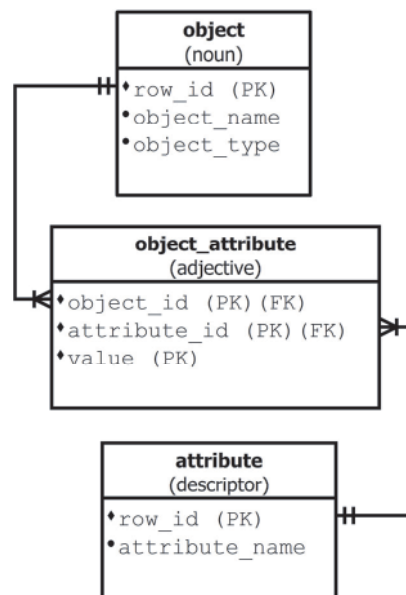


Рис. 2. Объектно-реляционный подход. Атрибуты объектов

возможным ввод информации структура которой не определена заранее, а изменение структурных связей типа «сущность-атрибут», «сущность-сущность» или «отношение-атрибут» может производиться во время работы приложения.

КЛАССИФИКАЦИЯ ДАННЫХ

Классификация [5] является наиболее простой и одновременно наиболее часто решаемой задачей универсальной модели хранения данных. Результат проведения классификации – это совокупность признаков, характеризующих группы исследуемых объектов как классы, к которым можно отнести новый объект, сохраняя общую концепцию, определяющую сходства и различия, а также другие особенности, уже классифицированной или вновь поступающей и анализируемой информации. По сути, классификацией является любое системное распределение объектов, явлений или понятий по каким-либо существенным признакам, выбранным для удобства их представления и обработки. Таким образом, классифицированная информация может быть представлена как упорядоченное по некоторому принципу множество абстрактных или конкретных сущностей, которые имеют сходные классификационные признаки (одно или несколько свойств), отобранные для определения критериев общности в описании, поведении или каких-либо других измерениях исходных неструктурированных данных. Обычно инструменты классификации разделяют по способу их воздействия на классифицируемый элемент информации. Например, возможна организация естественной классификации, которая может проводиться по существенным признакам, характеризующим общность предметов, понятий или явлений.

Предметная область, при этом, описывается элементарными единицами данных, соответственно относящихся к некоторым примитивам, перечень которых априори установлен (рис. 4). Основными понятиями при этом являются: «Мета-тип», «Тип объекта», «Объект».

Любой объект рассматривается как чистая абстракция [9], без привязки к какой-либо предметной области. Спецификация атрибутов объекта, согласно используемым объектно-ориентированным надстройкам для реляционной модели хранения, производится на уровне его типа, сущность «object_type». Семантика хранения предполагает, что каждый тип объекта наследует какому-либо базовому типу «meta_type», выраженному в терминах элементарных примитивов определяющих смысловой контекст как признак для возможной классификации всех дочерних элементов данных. Особое внимание нужно обратить на поле «parent_row_id» в описании сущности «object_type». Его предназначение, в создании древовидных иерархических структур, что можно рассматривать как эмуляцию множественного наследования.

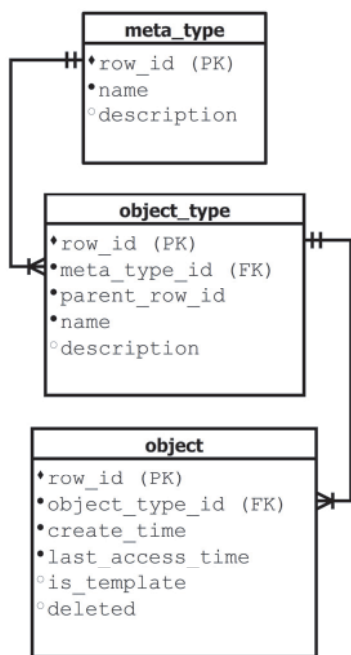


Рис. 4. Типизация данных. Фрагмент ER-модели

Также возможно применение искусственной или вспомогательной классификации, которая производится по внешнему или стороннему признаку и служит для придания множеству исследуемых элементов данных необходимого критерия упорядоченности. Такая классификация легко может быть выражена через категорирование, когда категории обеспечивают необходимый уровень косвенности, при определении критериев общности для элементов информации вне зависимости от их принадлежности к конкретному классу. Основными понятиями при этом становят-

ся: «Тип категории», «Категория», «Подчиненная категория» и «Присоединенная категория» (рис. 5). Любой объект может быть связан через свой тип с одной или несколькими категориями, что делает возможным при анализе или декомпозиции получение представления о его структуре и свойствах в терминах множества связанных с ним категорий (рис. 6). Каждая категория (сущность «category») наследует определенному типу (сущность «category_type»), являющему своеобразным дескриптором для иерархии категорий. Сами же иерархии могут быть созданы как древовидные структуры (поле «parent_row_id» в описании сущности «category»), что с точки зрения структурной целостности можно интерпретировать как эмуляцию композиции или агрегирования [5]. Присоединенные категории (сущность «linked_category») позволяют проводить ассоциацию между категориями, находящимися на разных уровнях иерархии или же устанавливать взаимосвязь с категориями, располагающимися в других, контекстуально связанных иерархиях.

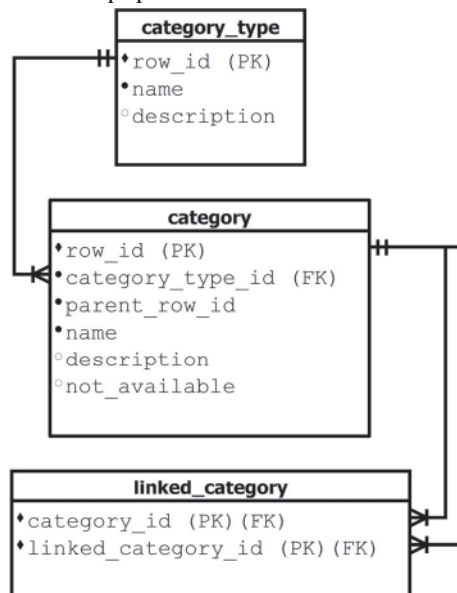


Рис. 5. Категорирование данных. Фрагмент ER-модели

Примером внешней классификации служит периодическая таблица химических элементов, где механизм категорирования может быть применен для указания принадлежности конкретного элемента к той или иной группе, например, щелочных металлов, галогенов и т.д.

Применение механизма типизации позволяет хранить информацию об объектах с подобным поведением и рассматривать их отношения как иерархию по принципу «от общего к частному». Применение категорирования позволяет хранить информацию об объектах, используя понятия агрегирования и композиции, а их отношения рассматривать по принципу «от целого к его части».

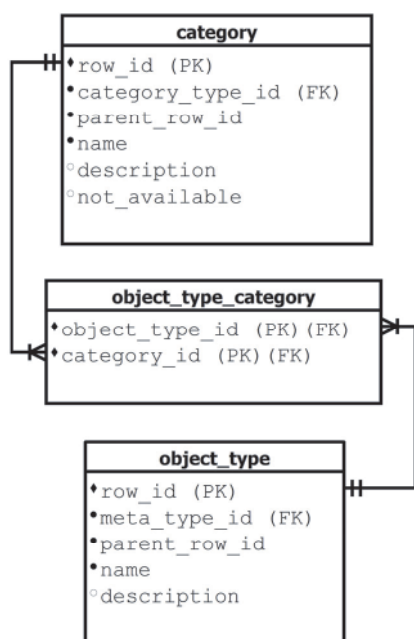


Рис. 6. Категорирование типов объектов.
Фрагмент ER-модели

Структура хранилища, построенного по принципам универсальной модели данных, логически может быть разделена на несколько основных взаимосвязанных составляющих. К ним относятся: инструменты классификации (типизация, категорирование), экземпляры и типы объектов (с возможностью назначения каждому типу определенного набора атрибутов), отношения объектов (позволяющие специфицировать типы отношений и значения соответствующих атрибутов). Все это делает возможным применение предлагаемой универсальной модели данных в различных областях научных исследований. Это могут быть задачи из области теоретической или экспериментальной химии связанные с конструированием и структурным анализом новых веществ, классификацией и систематизированным представлением данных о типах химических соединений, предсказания химических и физических свойств у кристаллических структур. Кроме того, во многих задачах из области физики предлагаемая универсальная модель может использоваться для хранения разнородной информации о результатах экспериментов с различным уровнем детализации или быть источником данных для построения математических моделей при проведении исследований. К таким задачам можно отнести анализ, распознавание и пространственную классификацию изображений, повышение чувствительности оптических датчиков и т.п.

ПРИМЕРЫ РЕАЛИЗАЦИИ

Далее приводятся примеры кода из DDL-скрипта для создания базы данных использующей универсальную модель хранения информации.

Пример кода создания таблицы мета-типов, т.е. набора элементарных примитивов, относящихся к некоторым внутренним понятиям предметной области, перечень которых априори установлен.

```

CREATE TABLE IF NOT EXISTS meta_type
(
    row_id uuid NOT NULL
        DEFAULT uuid_generate_v4(),
    name varchar(128) NOT NULL,
    description text,
    CONSTRAINT meta_type_pkey
        PRIMARY KEY(row_id),
    CONSTRAINT meta_type_uindex
        UNIQUE(alias)
);
  
```

Пример кода создания таблицы типов объектов, т.е. перечня концепций определяющих некоторую совокупность свойств, отличающих их владельца от других сущностей, не содержащих идентичный набор свойств.

```

CREATE TABLE IF NOT EXISTS object_type
(
    row_id uuid NOT NULL
        DEFAULT uuid_generate_v4(),
    meta_type_id uuid NOT NULL,
    parent_row_id uuid NOT NULL,
    name varchar(128) NOT NULL,
    description text,
    CONSTRAINT instance_type_pkey
        PRIMARY KEY(row_id),
    CONSTRAINT instance_type_fkey
        FOREIGN KEY(meta_type_id)
        REFERENCES meta_type(row_id),
    CONSTRAINT instance_type_uindex
        UNIQUE(alias)
);
  
```

Пример кода создания таблицы типов категорий, т.е. набора элементарных примитивов, относящихся к некоторым внешним понятиям предметной области, перечень которых априори установлен.

```

CREATE TABLE IF NOT EXISTS category_type
(
    row_id uuid NOT NULL
        DEFAULT uuid_generate_v4(),
    name varchar(128) NOT NULL,
    description text,
    CONSTRAINT category_type_pkey
        PRIMARY KEY(row_id),
    CONSTRAINT category_type_uindex
        UNIQUE(alias)
);
  
```

Пример кода создания таблицы категорий, т.е. инструмента обеспечивающего необходимую гибкость через дополнительный уровень косвенности при определении критериев общности для сущностей вне зависимости от их типизации.

```

CREATE TABLE IF NOT EXISTS category
(
    row_id uuid NOT NULL
        DEFAULT uuid_generate_v4(),
    category_type_id uuid NOT NULL,
    parent_row_id uuid NOT NULL,
    name varchar(128) NOT NULL,
    description text,
    not_available boolean,
    CONSTRAINT category_pkey
  
```

```

        PRIMARY KEY(row_id),
    CONSTRAINT category_fkey
        FOREIGN KEY(category_type_id)
        REFERENCES category_type(row_id),
    CONSTRAINT category_uindex
        UNIQUE(alias)
);

```

Пример кода создания таблицы присоединённых категорий, содержащей данные о внешних критериях общности по принципу отношений часть-целое.

```

CREATE TABLE IF NOT EXISTS linked_category
(
    category_id uuid NOT NULL,
    linked_category_id uuid NOT NULL,
    CONSTRAINT category_link_pkey
        PRIMARY KEY(category_id,
                    linked_category_id),
    CONSTRAINT category_link_fkey_1
        FOREIGN KEY(category_id)
        REFERENCES category(row_id),
    CONSTRAINT category_link_fkey_2
        FOREIGN KEY(linked_category_id)
        REFERENCES category(row_id)
);

```

Пример кода процедуры создания строки данных в таблице meta_type.

```

CREATE OR REPLACE FUNCTION create_meta_type_row
(
    meta_type_id uuid,
    name varchar(128),
    description text = NULL
)
RETURNS VOID AS
$BODY$
BEGIN
    INSERT INTO meta_type
    (
        row_id,
        name,
        description
    )
    VALUES(meta_type_id, $2, $3);
END;
$BODY$
LANGUAGE plpgsql;

```

Пример кода процедуры обновления данных в строке таблицы object_type.

```

CREATE OR REPLACE FUNCTION update_object_type_row
(
    object_type_id uuid,
    bit_mask integer,
    meta_type_id uuid = NULL,
    parent_row_id uuid = NULL,
    name varchar(128) = NULL,
    description text = NULL
)
RETURNS void AS
$BODY$
BEGIN
    UPDATE object_type t SET
    meta_type_id =
        CASE bit_mask & 1
            WHEN 1 THEN $3
            ELSE t.meta_type_id
        END,
    parent_row_id =
        CASE bit_mask & 2
            WHEN 2 THEN $4
            ELSE t.parent_row_id
        END,

```

```

    name =
        CASE bit_mask & 4
            WHEN 4 THEN $5
            ELSE t.name
        END,
    description =
        CASE bit_mask & 8
            WHEN 8 THEN $6
            ELSE t.description
        END
    WHERE t.row_id == object_type_id;
END;
$BODY$
LANGUAGE plpgsql;

```

Пример кода процедуры чтения данных из таблицы category_type.

```

CREATE OR REPLACE FUNCTION read_category_type_rows
(
    category_type_ids VARIADIC UUID[]
)
RETURNS SETOF record AS
$BODY$
BEGIN
    RETURN QUERY
    SELECT
        t.*
    FROM category_type t
    WHERE t.row_id == ANY(category_type_ids);
END;
$BODY$
LANGUAGE plpgsql;

```

Пример кода процедуры удаления строк данных из таблицы linked_category.

```

CREATE OR REPLACE FUNCTION
delete_linked_category_rows
(
    category_id uuid,
    use_linked boolean = NULL
)
RETURNS void AS
$BODY$
BEGIN
    DELETE FROM category_link t
    WHERE $1 =
        CASE COALESCE($2, false)
            WHEN true THEN
                t.linked_category_id
            ELSE t.category_id
        END;
END;
$BODY$
LANGUAGE plpgsql;

```

ЗАКЛЮЧЕНИЕ

Задача построения универсальной модели данных возникает при необходимости реализации системы хранения неструктурированной информации, с применением объектно-ориентированного способа ее представления. В процессе создания систем такого рода, наиболее подходящей основой является метод обеспечивающий декомпозицию сложных понятий до уровня элементарных примитивов. Использование таких примитивов является необходимой предпосылкой для разработки эффективного и надежного способа описания и извлечения информации при проведении исследований. В статье приводятся пояснения, из чего

состоят предлагаемые решения и методологии, основанные на фундаментальных понятиях программирования [6] и анализа данных. Основными преимуществами предлагаемого подхода можно считать возможность применения рассматриваемой универсальной модели к любому виду информации, а также возможность определения системы понятий [7], дающих основу для создания инструментов внутренней и внешней классификации данных. Наиболее приближенная к контексту конкретной сферы научных знаний, такая классификация может детально представлять взаимосвязь между структурой предметной области и тем, как в ней выражены ее общность и вариация.

СПИСОК ЛИТЕРАТУРЫ

1. *Silverstone L.* The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises. Wiley, 2001. 542 p.
2. *Silverstone L., Agnew P.* The data Model Resource Book, Vol. 3: Universal Patterns for Data Modeling. Wiley, 2009. 648 p.
3. *Фаулер М.* Архитектура корпоративных программных приложений. М.: Вильямс, 2007. 544 с.
4. *Эмблер С.В., Садаладж П.Дж.* Рефакторинг баз данных: Эволюционное проектирование. М.: Вильямс, 2007. 368 с.
5. *Booch G.* Object-Oriented Analysis and Design with Applications. Addison-Wesley, 2007. 720 p.
6. *Яблоков Д.Е.* Парадигмы программирования // XIV МНПК Научное обозрение физико-технических наук в XXI веке. 2015. №2(14). С. 94-98.
7. *Фаулер М.* Предметно-ориентированные языки программирования. М.: Вильямс, 2011. 577 с.
8. *Simsion G.C., Witt G.C.* Data Modeling Essentials, Third Edition. Morgan Kaufmann, 2005. 560 p.
9. *Hey D.C., Barker R.* Data Model Patterns: Conventions of Thought. Dorset House Publishing, 1996. 288 p.

UNIVERSAL DATA MODEL AS A CLASSIFICATION TOOL FOR SOLVING RESEARCH PROBLEMS

© 2016 D.E. Yablokov

Samara National Research University named after Academician S.P. Korolyov

The article describes the concept of a universal data model for applications, which oriented to experts who are participating in scientific research. Definitions and examples of using mechanisms of typification and categorization as tools of internal and external classification based on internal and external properties of objects are given. It is supposed that the storage structure will not be strongly bound to any scientific domain, and it will be demanded in many projects connected to accumulation, the analysis and data processing. The proposed approach to the organization of storage system is based on the main principles, concepts and technologies used in the software design process. These include an analysis of the data domain and finding commonality and variability of criteria, which are a guarantee of a good abstraction technique, which is one of the basic tools for working with data.

Keywords: universal data model, storage structure, the abstraction level, data typification, internal classification, data categorization, external classification.