Паттерны проектирования моделей баз данных как систем хранения экспериментальной информации при решении исследовательских задач

Д.Е. Яблоков

Самарский национальный исследовательский университет имени академика С.П. Королёва, 443086, Мсоковское шоссе, 34, Самара, Россия

Аннотация

В статье рассматриваются паттерны проектирования реляционных баз данных, используемых в качестве систем хранения экспериментальной информации. Приводится классификация таких паттернов, основываясь на их сложности и уровне детализации при описании сущностей и их отношений. Для каждого паттерна показаны специфические особенности его применения в процессе моделирования данных. Базы данных, построенные на основе простых паттернов, менее приспособлены к изменениям. Их дизайн соответствует только контексту решаемой задачи. Базы данных, созданные с использованием более сложных паттернов, имеют более гибкий дизайн и учитывают требования, которые могут возникнуть в будущем, что сводит к минимуму необходимость их перепроектирования.

Ключевые слова: модель данных; паттерн проектирования; декларативный паттерн; расширенный декларативный паттерн; контекстуальный паттерн; типизированный контекстуальный паттерн; расширенный контекстуальный паттерн

1. Введение

Важным обстоятельством, при проектировании систем хранения экспериментальной информации, является проблема правильного выбора стратегии моделирования данных, т.е. выбора основной концепции, которая позволяла бы отобразить их информационное содержание и была бы достаточно гибкой с точки зрения возможных расширений [1]. Под концепцией будем понимать основанную на целостных и систематизированных представлениях совокупность взглядов, позволяющих выражать определенный способ понимания или трактовки каких-либо предметов, событий, процессов или явлений, представляющих ту или иную информационную ценность. Следуя этому определению можно сказать, что моделирование данных, как собственно и сам выбор модели данных, является очень важным этапом, в процессе разработки базы данных, закладывающим основы понятийного аппарата, в терминах которого будет производиться работа с системой хранения. Конкретная модель данных, при этом, является своего рода паттерном [2], т.е. зафиксированным воспроизводимым средством описания способа представления и хранения информации с учетом выбранного уровня абстракции, связанного с контекстом предметной области, которая в дальнейшем будет являться источником данных.

2. Декларативный паттерн

При таком подходе [3] описание данных производится по принципу «как есть» (рис. 1). Многие вычислительные приложения для экспериментальных исследований часто используют некоторый набор элементов взаимосвязанных между собой определенным набором соединений. Например, экземпляр какой-либо абстрактной структуры данных «graph» может содержать некоторый набор сущностей, обладающих семантикой поведения вершины графа «vertex». Пусть принадлежность этих сущностей к обозначенной выше структуре данных, описывается с помощью вспомогательного понятия «graph_vertex». И пусть они могут объединяться в пары с помощью связей «edge», ассоциативных сущностей, обладающих семантикой поведения ребра графа. Вершины и ребра могут представлять собой объекты любой природы [4], которые, как правило, имеют в своем описании какую-либо характеристику, позволяющую идентифицировать их среди множества подобных объектов. Кроме того, они могут быть снабжены и некоторыми дополнительными атрибутами, касающимися, например, положения или статуса вершины, веса или ориентированности ребра, а также сведениями о свойствах той абстракции, которая в данный момент может рассматриваться как вершина или ребро. Наряду со свойствами, напрямую относящимися к таким автономным понятиям как граф и вершина, паттерн предполагает атрибутирование отношений, т.е. соответствующие наборы атрибутов ассоциативных сущностей «graph_vertex» и «edge».

Внешние ключевые атрибуты отношений для связей между вершинами («from_vertex_id» и «to_vertex_id») указывают их направление в случае представления данных в виде планарного или пространственного ориентированного графа. В случае неориентированного графа информация о связи должна дублироваться только в обратном направлении, таким образом, чтобы значения атрибутов «from_vertex_id» и «to_vertex_id» в строке, содержащей информацию об обратной связи, поменялись местами. Это позволит представлять данные о связях между вершинами простого неориентированного графа в терминах параллельных ребер ориентированного мультиграфа. Из-за того, что данный паттерн является узкоспециализированным решением, можно провести однозначное соответствие между объектами предметной области и абстракциями, используемыми в процессе моделирования.

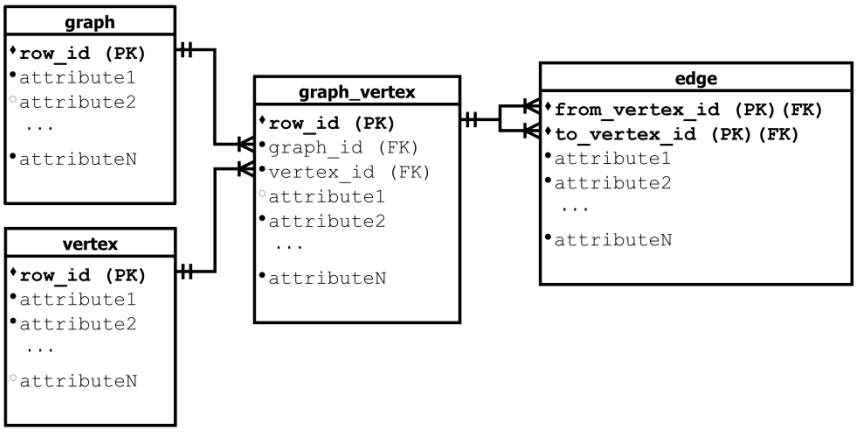

Рис. 1. Декларативный паттерн.

Достоинства паттерна:

1. Легкость в понимании. Наличие небольшого набора абстракций позволяет сравнительно просто моделировать предметную область, что предполагает точные формальные определения, которые интуитивно понятны.
2. Легкость в сопровождении. Возможность манипулирования данными без необходимости знания особенностей более развитой структуры хранения.
3. Относительная простота запросов при выборке данных. Наличие простого и в то же время мощного аппарата извлечения данных, опирающегося главным образом на теорию множеств и математическую логику.
4. Высокая скорость реализации системы доступа к данным, а также минимальные трудозатраты на ее сопровождение.

Недостатки паттерна:

1. Зафиксированная структура хранения. Требуется изменение структуры хранения для добавления новой сущности или атрибута. Для этого нужно будет вводить в действие новую таблицу, как абстракцию для описания некоторого объекта предметной области или атрибут, как аналог какого-либо его свойства. Это требует проведения операций рефакторинга типа «введение новой таблицы» или «введение нового столбца» [5] для уже существующей базы данных, которые подразумевают кроме процедуры обновления схемы еще и необходимость заполнения созданных структурных единиц новыми данными или переноса в них уже существующих данных, а также обновление всех программ доступа. В случае отсутствия или слабой проработки стратегии адаптации системы хранения к поступающим требованиям, затрагивающим содержание или качество уже присутствующей или вновь поступающей информации подобные действия, могут привести к негативным последствиям, связанным с вопросами информационной или функциональной семантики хранения.
2. Неуниверсальность и слабый уровень абстракции. Ограниченность знаний о специфике предметной области (прямое следствие простоты реализации) и невозможность адекватного отражения ее семантики. Атрибуты или связи между сущностями могут быть выражены только в терминах выбранного понятийного аппарата. Ситуация подразумевающая ввод нового понятия в систему хранения является неблагоприятной, т.к. данные касающиеся, расширения контекста описываемой предметной области, очень сложно логически встроить в уже имеющийся понятийный аппарат. В большинстве случаев приходится задумываться о целесообразности подобных модификаций т.к. объемы трудозатрат для адаптации существующей базы данных к специфицированным требованиям и создания новой системы хранения находятся примерно на одном уровне.

Использование паттерна данного типа оправдано, если изменение контекста предметной области не предполагается в будущем, а развитие структуры хранения будет проводиться за счет ввода подчиненных сущностей, характеризующих дополнительную информацию об объектах уже присутствующих в базе данных.

3. Расширенный декларативный паттерн

Имеет структуру подобную предыдущему паттерну, но реализует дополнительный уровень косвенности при описании атрибутов и их значений. Например, модель данных для хранения информации из области кристаллохимии с использованием уже описанных графовых абстракций может выглядеть следующим образом (рис. 2). Внешние ключевые атрибуты «unit_cell_id» и «space_group_id» сущности «structure» специфицируют ее связи с сущностями «unit_cell» и «space_group» описывающими понятия элементарной ячейки [4] и пространственной группы [4] соответственно. Эти данные необходимы для однозначной уникальной идентификации экземпляра химической структуры с точки зрения кристаллохимии. Атрибуты «type_symbol» и «atomic_number» сущности «atom» описывают основные характеристики химических элементов из периодической таблицы, т.е. тип элемента и его атомный номер. Атрибут «value», содержащийся в описании ассоциативных сущностей «structure_data», «atom_data», «structure_atom» и «bond» необходим для сохранения значений, связанных с этими сущностями свойств. Сами же свойства, как информационное отображение возможных характеристик объектов предметной области, описываются с помощью

структурных единиц «structure_attribute», «atom_attribute», «structure_atom_attribute», «bond_attribute» содержащих идентичный набор полей «name», «type» и «size» представляющих имя, тип и размер, определяющий диапазон значения конкретного свойства.

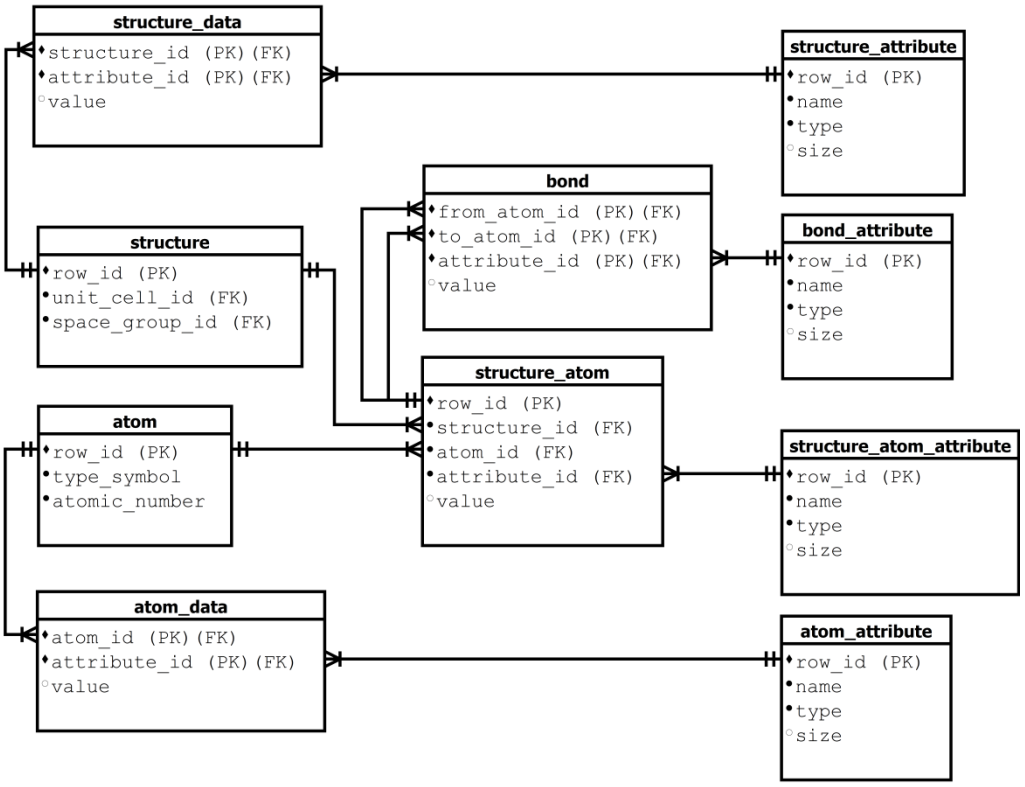

Рис. 2. Расширенный декларативный паттерн.

Данный паттерн включает все достоинства декларативного подхода и обладает более развитым механизмом представления атрибутов, что позволяет без серьезных ограничений описывать различные состояния для экземпляров сущностей определенных классов и их отношений. Если с течением времени, обнаруживаются новые данные, касающиеся каких-либо характеристик, состояний или измерений структур, атомов и их отношений, данный паттерн позволяет сохранить эту информацию без перепроектирования системы в целом или какой-то конкретной ее части. Кроме того, в приложениях доступа к данным, слой бизнес-логики может быть организован с большим заделом гибкости и масштабируемости из-за возможности удачного сочетания сравнительно простой схемы данных и удобной процедурной модели доступа, определяющей четкие границы в процессе реализации.

Но, при меньшей фиксации структуры хранения касающейся описательной части атрибутов и их значений, ввод в эксплуатацию новой сущности невозможен без перепроектирования всей системы хранения или конкретной ее части. Описания некоторых атрибутов могут повторяться для различных классов сущностей, а это говорит об избыточности данных об атрибутах. Вследствие избыточности возможно обновление описания атрибута только для одного класса сущностей. В такой ситуации база данных будет содержать различные описания для идентичных атрибутов, а это потенциальная противоречивость при обработке или обновлении данных. На уровне приложения доступа к данным должны быть реализованы операции преобразования типов для конвертации значений атрибутов к типу, указанному в описании атрибута.

Сценарий, иллюстрирующий возможность применения данного паттерна может быть следующим. Если количество абстракций, используемых в процессе моделирования для описания объектов предметной области, остается неизменным, но информация, связанная с их свойствами, постоянно меняется, то это является рекомендацией для того чтобы воспользоваться данным паттерном. Основной акцент необходимо сделать на том, что если эти изменения затрагивают сигнатуру свойств какого-либо объекта, т.е. их имя и тип, то применение расширенного декларативного паттерна не только оправдано, но и необходимо.

4. Контекстуальный паттерн

Данный паттерн [3] подразумевает независимость от информационного контекста, когда различные объекты могут рассматриваться по-разному в зависимости от ситуации, причем изменение или назначение им состояния или поведения может производиться даже в режиме работы приложения. Использование ассоциативных сущностей, формализующих взаимосвязь между объектами и атрибутами, а также связь между атрибутами и отношениями сущностей позволяет назначать любому объекту или отношению объектов произвольное, продиктованное контекстом решаемой задачи, количество атрибутов. Общая диаграмма контекстуального паттерна (рис. 3) и его упрощенная модель данных (рис. 4) предполагают, что описание любой предметной области вне зависимости от контекста может быть выражено в терминах объектов, их атрибутов, отношений объектов и атрибутов отношений.

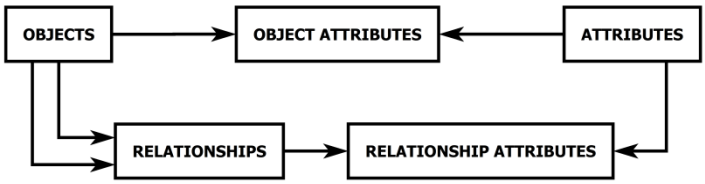

Рис. 3. Общая диаграмма контекстуального паттерна.

В описании каждого объекта присутствуют поля «name» и «description», отвечающие за смысловой контекст его экземпляра. Для каждого атрибута предполагается его формальное описание, содержащее имя («name»), тип данных («type») и размер («size»). Фактически, таким образом, происходит объявление атрибутов без указания фактических значений. Значения атрибутов, связанных с конкретным экземпляром объекта или объектами, могут быть определены с использованием поля «value» ассоциативной сущности «object_attribute». Таким же образом значения атрибутов для экземпляров сущностей «relationship» определяются в поле «value» ассоциативной сущности «relationship_attribute». Семантика такого подхода очень похожа на расширенный декларативный паттерн, но в рамках контекстуального паттерна осуществляется попытка ухода от контекста конкретной предметной области.

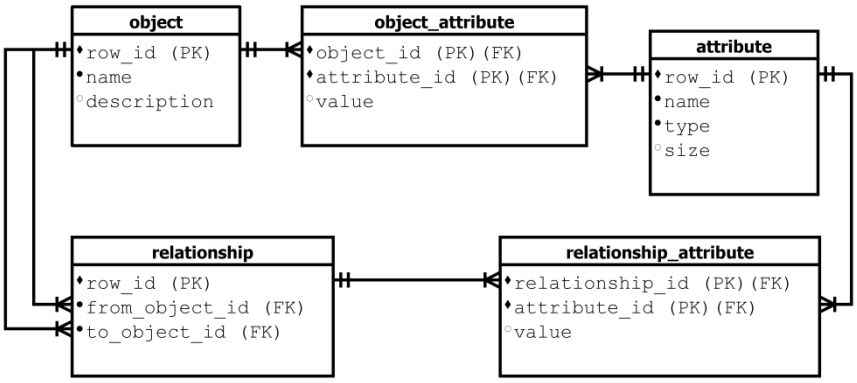

Рис. 4. Контекстуальный паттерн.

Данный паттерн относительно легко использовать при соблюдении определенного набора неявно выраженных правил, но, за дополнительный уровень гибкости структуры хранения приходится платить. Из-за потенциальной возможности изменения стратегии работы с данными могут возникнуть сложности с пониманием семантики хранения и выборкой данных. Структура хранения, не ориентированная на логику предметной области, как минимум, потребует дополнительных средств доступа к данным в виде набора пользовательских функций или представлений, упрощающих нестандартный доступ к хранящейся в базе данных информации. Отсутствие четкой спецификации принадлежности конкретного объекта к определенному сегменту предметной области может привести к снижению производительности за счет выборки большого числа экземпляров объектов с идентичным набором атрибутов. Необходимость хранения информации обо всех атрибутах в одном месте может привести к проблеме избыточности данных об атрибутах, описанной в недостатках расширенного декларативного паттерна.

5. Типизированный контекстуальный паттерн

Решает проблему отсутствия в описании объектов признака принадлежности каждого из них определенному классу, зависящему от контекста предметной области. За счет ввода системы пользовательских типов для объектов и их отношений данный паттерн позволяет классифицировать данные о них по заранее не определенным признакам (рис. 5). Появление таких признаков в системе хранения может производиться поэтапно, например, при формировании определенного уровня понимания особенностей предметной области, которые не были известны на первоначальном этапе работы с базой данных. Перенос информации о типах атрибутов в отдельную структуру хранения позволяет решить проблему избыточности данных.

Структура универсального хранилища [6], построенного на основе принципов типизированного контекстуального паттерна, концептуально может быть разделена на несколько основных взаимосвязанных составляющих. К ним относятся:

1. Объекты, объединяющие такие понятия как тип объекта («object_type») и экземпляр объекта («object») [7].
2. Атрибуты («attribute») и типы значений атрибутов («data_type»), предполагающие возможность отдельного описания сигнатур свойств объектов [7].
3. Отношения объектов, ассоциативно связанные с типами связей объектов («relationship_type») для формализации класса сущности, представляющей собой отношение («relationship»).
4. Атрибуты объектов («object_attribute») и атрибуты отношений («relationship_attribute») имеющие возможность хранения значений атрибутов объектов и их отношений соответственно.

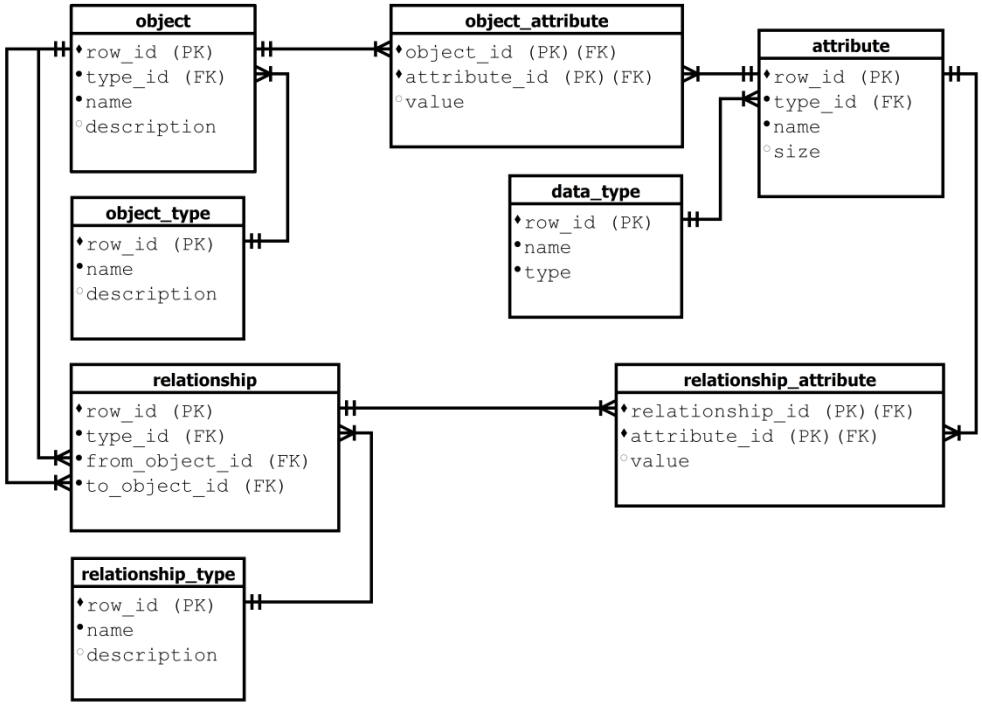

Рис. 5. Типизированный контекстуальный паттерн.

К сожалению, применение типизированного контекстуального паттерна осложняется одним из ключевых недостатков расширенного декларативного паттерна. Операции конвертации, которые необходимо проводить для получения значений атрибутов сообразно их типам по-прежнему должны быть реализованы в тексте запросов или представлений, либо в приложении, на уровне слоя доступа к данным.

6. Расширенный контекстуальный паттерн

В базе данных, созданной по принципу расширенного контекстуального паттерна, важно определить общие для большинства потребителей данных примитивы, основанные на априори определенных элементарных понятиях. Это позволит однозначно идентифицировать логически связанные с этими понятиями неструктурированные данные вне зависимости от предметной области и обеспечит переносимость модели данных из проекта в проект. При этом отпадет необходимость каждый раз модифицировать старую или разрабатывать новую схему хранения и проводить кардинальные изменения программ, которые взаимодействуют с базой данных. Степень детализации, при использовании такого подхода, обязывает разработчиков новых приложений продумывать на ранних этапах проектирования лишь наиболее важные вопросы, поскольку все необходимые нюансы развития системы хранения могут быть учтены позже, когда концептуальное преставление о модели данных, по мере углубления знаний, касающихся проблемных участков предметной области, разовьется до необходимого уровня.

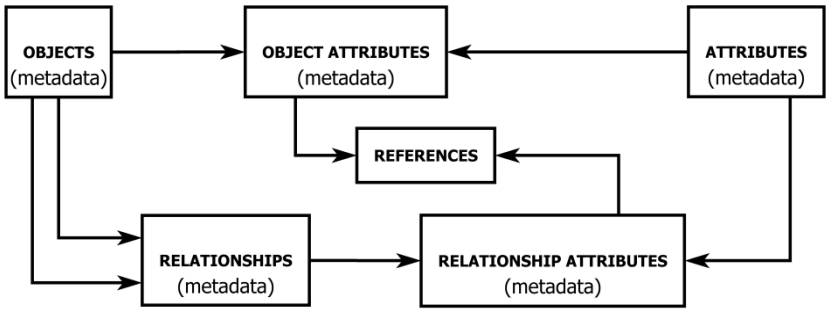

Рис. 6. Общая диаграмма расширенного контекстуального паттерна.

Ввод дополнительного уровня метаданных позволил избавиться от недостатков присущих декларативным и контекстуальным паттернам и заложить в расширенный контекстуальный паттерн все необходимые особенности для соответствия требованиям, предъявляемым к системам хранения, работающим на основе универсальной модели данных. К ним относятся:

1. Возможность представления объектов любой предметной области.
2. Легкость понимания, как специалистами, так и простыми пользователями. Возможность предоставления данных с использованием элементов предметно-ориентированного языка.

3. Отсутствие избыточности данных и операционная полнота. Исключение излишней информации и поддержка всех возможных операций над данными.
4. Способность эволюционировать с целью включения новых требований с минимальным влиянием на уже существующие данные.

Описание данных производится с использованием реляционного подхода и элементов из теории объектно-ориентированного программирования. Основной акцент делается на том, что при сохранении реляционного ядра 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 и композиция которые позволяют представлять отношения между сущностями по принципам «подобного поведения» или «является частью» соответственно. Объектно-ориентированный подход позволяет представлять данные в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром сущности определенного класса. Это способствует правильному и более эффективному структурированию хранимой информации, а также делает возможным проведение объектно-ориентированной декомпозиции при формировании концептуальных границ модели данных.

Как и в случае с типизированным контекстуальным паттерном структура хранилища, построенного по принципам расширенного контекстуального паттерна, может быть разделена на несколько составляющих. Далее приводится краткое описание этих составляющих, и даются пояснения к применению некоторых категорий идей, на базе которых построены предлагаемые решения и методологии работы с данными.

Объекты. Любой объект, как экземпляр сущности определенного класса, рассматривается как чистая абстракция, без привязки к какой-либо предметной области. Спецификация свойств объектов, согласно используемым объектно-ориентированным надстройкам для реляционной модели хранения, производится на уровне типа объекта, сущность «object_type» (рис. 7). 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 в семантике хранения расширенного контекстуального паттерна означает эмуляцию наследования.

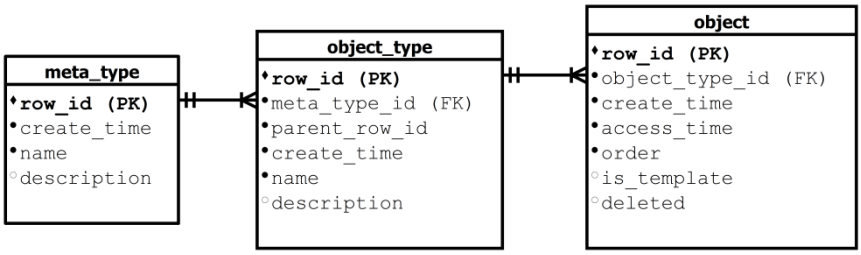

Рис. 7. Расширенный концептуальный паттерн. Объекты.

Отношения. Каждому экземпляру отношения между объектами («relationship») ставится в соответствие определенный экземпляр объекта («object»), задавая таким образом смысловую принадлежность того или иного отношения к конкретному классу (рис. 8). Например, тип связи между атомами («object_type») может быть обозначен как «HB» (Hydrogen Bond – водородная связь), а базовым типом («meta_type») для данного типа связи будет являться мета-тип «Edge» (ребро). Это дает гарантию, что при проведении анализа или декомпозиции этого межатомного отношения его можно будет рассматривать в терминах примитивных графовых абстракций. Кроме того, описание каждого отношения между объектами снабжено соответствующим типом отношения («relationship_type»), определяющим необходимый уровень абстракции, для выделения существенных характеристик, отличающих конкретный экземпляр отношения от отношений, принадлежащих другим классам. Например, тип отношения может содержать информацию, касающуюся взаимоотношений объектов по принципу «часть-целое», т.е. специфицировать форму отношения между объектами по таким уровням как осведомленность, агрегирование или композиция. В процессе проведения эксперимента для описания иерархических данных очень часто возникает необходимость их определения не от родителя к предку, а в обратном порядке, когда изначально специфицируются дочерние элементы, а уж потом их владелец. Ассоциативная сущность «object_relationship» позволяет указывать взаимосвязь между дочерними и родительскими элементами вне зависимости от их жизненного цикла, например, создавать подмножество элементов графа только с помощью ребер или только с использованием вершин.

Атрибуты. Каждый атрибут («attribute») содержит в своем описании информацию о структурном коде (поле «structural_code»), задающим семантику его хранения. Это могут быть как примитивные атрибуты (рис. 9), соответствующие определенным типам данных, так и составные, состоящие из примитивных или таких же составных атрибутов. Ассоциативная сущность «attribute_data_type» кроме ссылки на соответствующий тип данных (поле «data_type_id»), определяющего сигнатуру примитивного атрибута, содержит также поле «attribute_exid», которое используется для конструирования псевдонима таблицы, в которой будут храниться значения примитивных атрибутов данного типа.

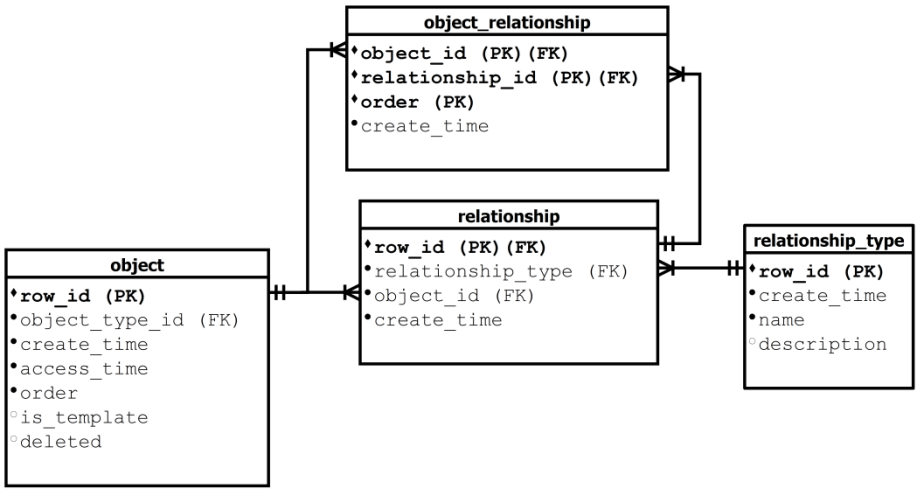

Рис. 8. Расширенный концептуальный паттерн. Отношения.

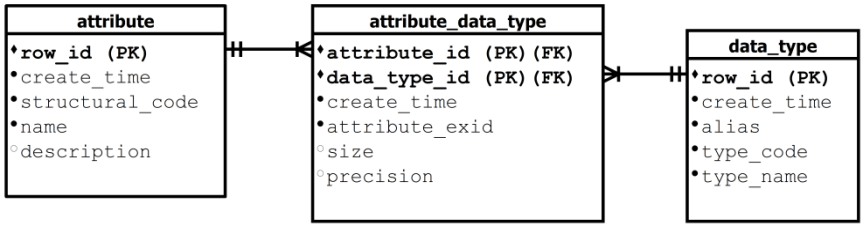

Рис. 9. Фрагмент инфологической модели. Атрибуты.

Атрибуты объектов и отношений. Каждому типу объекта («object_type») назначается определенное количество атрибутов (рис. 10). Ассоциативная сущность «object_type_attribute» с помощью поля «parent_row_id» позволяет производить уточнение, какому конкретному составному атрибуту принадлежат те или иные атрибуты, являющиеся по отношению к нему дочерними. Предполагается, что по умолчанию спецификатор доступа к свойствам объекта-предка всегда будет public. Это означает, что в процессе описания иерархий объектов следует придерживаться подстановочного принципа Барбары Лисков, являющегося одним из основных принципов для работы в объектно-ориентированном стиле. Исключение определенных атрибутов из области видимости типа-потомка возможно при условии, когда значение поля «not_inherited» будет установлено в «true», т.е. определен уровень доступа private.

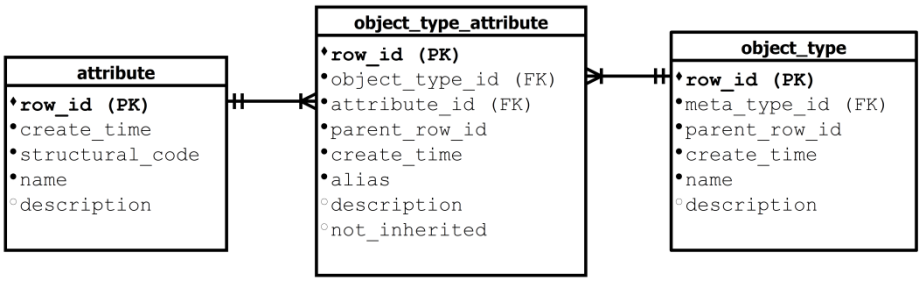

Рис. 10. Фрагмент инфологической модели. Атрибуты объектов и отношений объектов.

Значения свойств. Сохранение значений примитивных атрибутов производится в отдельных таблицах (например, «attribute_value_18F19DCCA4F24B27B3D0BAB50AAE740B»), что позволяет во время конструирования запросов не заботиться об их конвертации (рис. 11). Причем псевдонимы таких таблиц содержат в качестве постфикса идентификатор (поле «attribute_exid»), объявленный на уровне ассоциативной связки («attribute_data_type») для типов данных и атрибутов, что и определяет тип сохраняемых значений.

Ссылки. Введенный механизм ссылок («attribute_reference») на атрибуты объектов позволяет определять относительное значение для другого атрибута объекта или атрибута отношения между объектами (рис. 12). Такая же функциональность распространяется и на атрибуты отношений объектов, ссылки на которые могут использоваться в качестве относительных значений для других атрибутов отношений или атрибутов самих объектов. В случае возникновения необходимости организации справочников, содержащих какую-либо служебную информацию или же набор констант, используемых при описании экспериментальных данных, ссылки на свойства записей этих справочников можно использовать как относительные значения для атрибутов других элементов данных. Это избавит систему хранения от дублирования информации, т.е. сделает данные более нормализованными, а также предотвратит возникновение проблем, связанных с ссылочной целостностью.

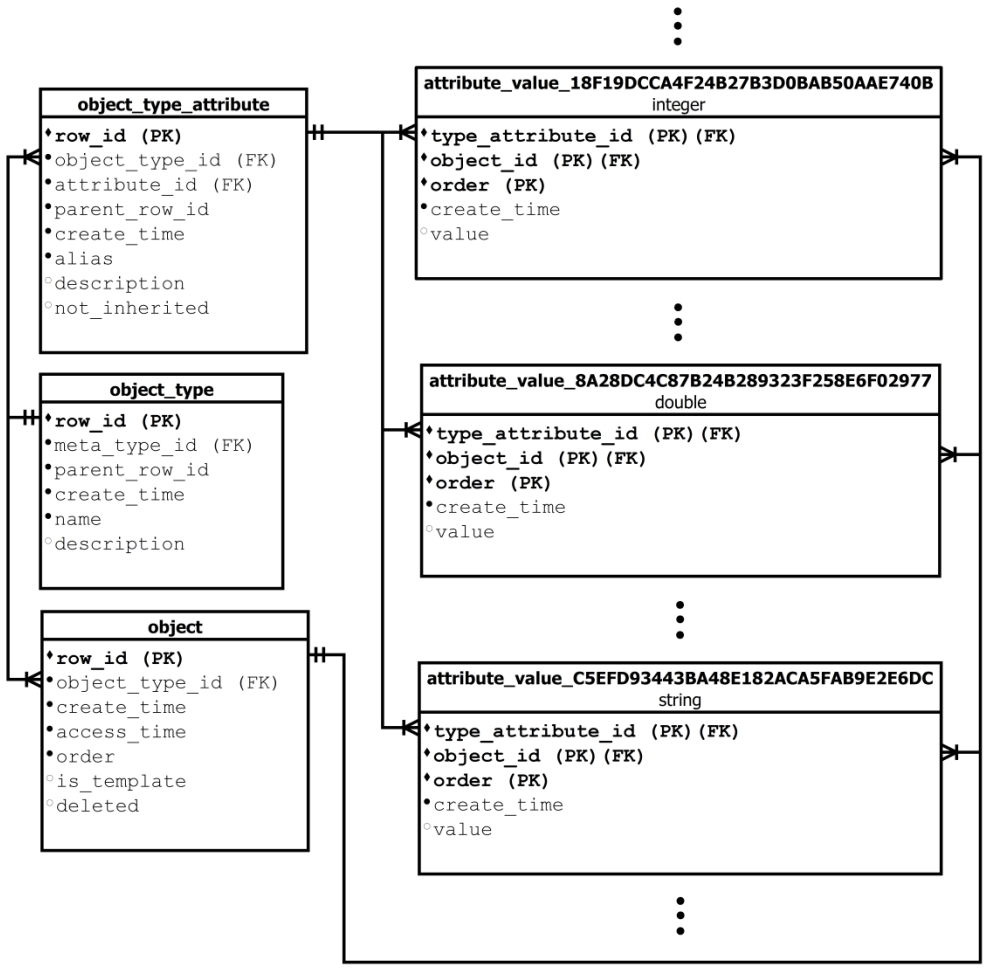

Рис. 11. Фрагмент инфологической модели. Значения свойств.

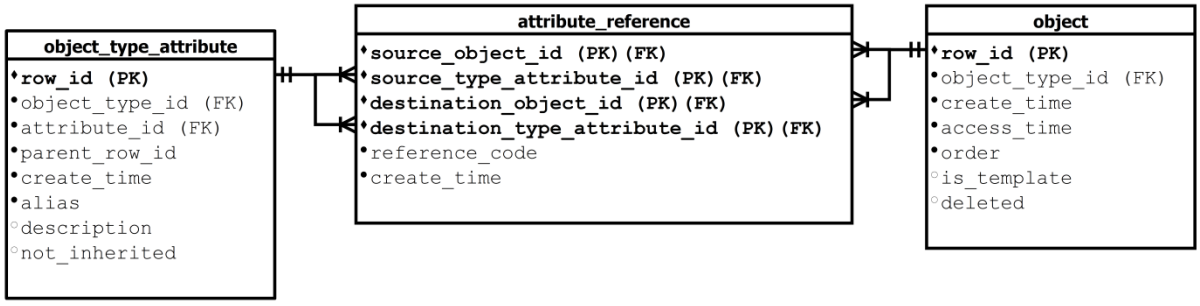

Рис. 12. Фрагмент инфологической модели. Ссылки на свойства.

7. Заключение

Выбор соответствующиего паттерна модели данных – это по сути процесс выбора парадигмы работы с данными на основе модели предметной области и используемого уровня абстракции. Ценность применения любого из паттернов состит в том, что освоившись с подходом, мы получаем в свое распоряжение множество приемов, в соответствие уровню сложности паттерна, позволяющих контролировать как сложность предметной области так и сложность системы хранения, а также программного обеспечения для работы с данными. Разумеется каким бы ни был выбранный подход он не заменит наличие опыта и стиля мышления необходимого для правильного выбора стратегии работы над соответствующим проектом обработки и анализа данных.

Благодарности

Работа по анализу и классификации паттернов моделей данных выполнена при поддержке правительства Российской Федерации (грант 14.B25.31.0005).

Литература

[1] Simsion, G.C., Witt, G.C. Data Modeling Essentials, Third Edition / Graeme C. Simsion, Graham C. Witt – Morgan Kaufmann Publishers, 2005. – 560 p.

[2] Hey, D.C. Data Model Patterns: Conventions of Thought / David C. Hey – Dorset House Publishing, 1996. – 288 p.
[3] Silverstone, L. The data Model Resource Book, Vol. 3: Universal Patterns for Data Modeling / Len Silverston – Wiley Computer Publishing, 2009. – 648 p.
[4] Blatov, V.A., Proserpio, D.M. Periodic-Graph Approaches in Crystal Structure Prediction / edited by A. R. Oganov // Modern Methods of Cristal Structure Prediction. – Wiley-VCH, 2011. – P. 1-28.
[5] Ambler, S. Refactoring Databases: Evolutionary Database Design / Scott W. Ambler, Premodkumar J. Sadalage – Addison-Wesley, 2006. – 384 p.
[6] Silverstone, L. The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises / Len Silverston – Wiley Computer Publishing, 2001. – 542 p.
[7] Fowler, M. Patterns of Enterprise Application Architecture / Martin Fowler – Addison-Weatley, 2003. – 736 p.