электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 681.3.07

Процесс поиска отображений при интеграции схем данных

***Белошицкий Д.А.**, аспирант,*
Кафедра «Компьютерные системы и сети»
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

Научный руководитель: Брешенков А.В., д.т.н., профессор
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
v.suzev@bmstu.ru

Введение

В настоящее время системы управления базами данных (СУБД) являются наиболее распространенным средством хранения данных и обеспечения доступа к ним. Модель данных, получившая наибольшее распространение в современных СУБД – это реляционная модель данных. Ее принципы были впервые сформулированы Э.Ф. Коддом в статье «A Relational Model of Data for Large Shared Data Banks», ставшей классической.

Современные СУБД предлагают разработчику набор взаимосвязанных сервисов, которые позволяют абстрагироваться от проблем хранения, управления данными, обеспечения целостности, производительности и безопасности, позволяя таким образом сосредоточиться на специфичных задачах приложения, которое использует базу данных.

К сожалению, в настоящий момент довольно редко возникают ситуации, когда все данные организации могут быть помещены в единую базу данных: все чаще и чаще разработчики информационных систем сталкиваются с множеством слабо связанных и разрозненных источников информации и им приходится постоянно выполнять рутинные действия по выполнению низкоуровневых операций с данными в гетерогенной среде.

Все это влечет за собой появление новых задач, которые должны решаться в контексте гетерогенной информационной среды организации. Среди них следует выделить следующие: поиск источников информации, семантическая обработка запросов (источник данных, к которому обращен запрос, должен определяться в момент run-time, пользователя необходимо освободить от задачи выбора наиболее актуального и достоверного источника данных), построение правил и ограничений целостности для запросов, обрабатывающих несколько источников данных одновременно, ограничение доступа, журналирование, работа с метаданными.

Особое место в этом списке занимают вопросы интеграция данных, которые, вообще говоря, могут быть представленными различными моделями. Например, необходимо объединить набор реляционных таблиц и xml-файл или реляционную и объектно-ориентированную БД.

В данной работе внимание уделено вопросам интеграции схем данных, представленных реляционной моделью, ввиду ее наибольшей распространенности. Первые публикации в этой области появились в 90-х годах прошлого века . Несомненный интерес представляет работа, являющаяся обзором методик интеграции того времени.

Строго говоря, понятие «Интеграция схем данных» не совсем точное, так как схему данных неправильно рассматривать в отрыве от контекста самих данных. С развитием machine learning и information retrieval стало возможным анализировать семантику данных. Благодаря учету смыслового содержания данных их интеграцию можно проводить более качественно. Поэтому правильнее говорить об интеграции данных, а не только их схем.

Фундаментальной и базовой операцией в области интеграции схем данных является операция *Match* (англ. совпадать). На вход этой операции поступают две схемы данных, а на выходе формируется отображение между элементами входных схем данных, которые семантические соответствуют друг другу. Далее будем называть процесс получения отображения термином «мэтчинг» (от англ. matching). В настоящее время мэтчинг выполняется вручную с использованием графического интерфейса. К недостаткам данного процесса можно отнести его рутинность, длительность и большое количество ошибок. Учитывая огромные размеры схем современных баз данных и тенденцию к их роста, можно сделать вывод о том, что в ближайшем будущем процесс мэтчинга будет не просто затруднен, а попросту невозможен. В самом простом случае, когда вычислительная сложность операции *Match* - *O(n)*, на практике же необходимо находить максимально близкие по семантике элементы входных схем – в этом случае вычислительная сложность будет резко возрастать. Поэтому необходимо автоматизировать данный процесс.

Проблемы интеграции данных

Базовые проблемы, с которыми приходится сталкиваться при интеграции схем данных начинаются со структурных различий в схемах данных и заканчиваются семантическими. Далее приведена классификация основных типов проблем, которые имеют место.

Различные точки зрения проектировщиков

Во время проектирования схемы данных различные проектировщики отображают свои субъективные представления о предметной области в схему данных проектируемой БД. В результате этого появляются концептуальные различия в интегрируемых схемах данных. Ситуацию поясняет простой пример, показанный на рисунке ниже.

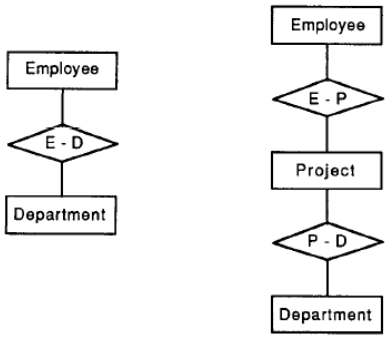

Рис. 1. Концептуальные различия в схемах данных

На рисунке изображены две простые схемы данных, которые хранят в себе информацию об организационной структуре организации. В схеме слева сущность Employee связана с сущностью Department отношением E-D. Таким образом, сущность Employee принадлежит сущности Department. На схеме справа эти же сущности связаны между собой через сущность Project.

Эквивалентность

При рассмотрении схем баз данных, моделирующих одну и ту же предметную область, часто могут быть обнаружены части, которые эквивалентны, но состав сущностей, атрибутов и отношений при этом различен. На рисунке ниже изображены две две эквивалентные схемы данных.

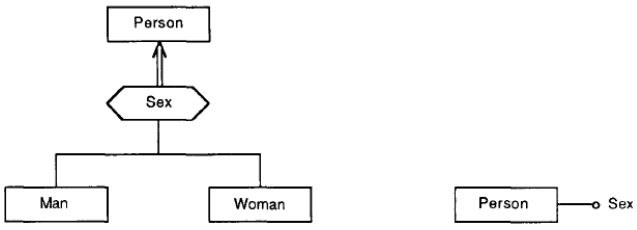

Рис. 2. Эквивалентные схемы данных

Сущности Man и Woman в схеме слева представлены как члены иерархии сущностей, тогда как на схеме справа они выражены через атрибут.

Различные спецификации

У проектировщика БД есть огромный выбор при проектировании схемы данных: можно использовать различные названия, нотации, типы данных, ограничения целостности, зависимости; можно создавать различные триггеры, которые добавляют

поведенческий аспект в схему данных. При анализе названий могут проявляться эффекты синонимии, омонимии и т.д..

Различная семантика

Сущности с аналогичным набором атрибутов, аналогичными названиями и ограничениями целостности могут быть отображениями не идентичных объектов реальности. Например, сущность Компания в одной схеме может означать только торговую компанию, тогда как в другой схеме сущность с тем же именем может означать только IT-компанию. В этом случае нужен лингвистический анализ и словарь данных, в котором находится классификация различных слов по областям их применения и т.д. В приведенном примере сущности являются абсолютно разными и их не нужно интегрировать.

Мощность отображения (мэтчинга)

Далеко не всегда получается для каждого элемента одной схемы найти один единственный соответствующий по смыслу элемент другой схемы. Довольно часто возникают ситуации, когда элементу одной схемы можно поставить в соответствие несколько элементов-кандидатов другой схемы или несколько элементов, композиция которых соответствует элементу первой схемы. Для описания и количественной оценки данного явления вводят такое понятие, как мощность отображения (мэтчинга) [5]. Пусть S_1 и S_2 – две входные схемы. Пусть элементу s1 схемы S_1 после выполнения операции Match (S_1, S_2) поставлено в соответствие n элементов-кандидатов из схемы S_2. В этом случае мощность отображения 1:n. Мощность может быть выражена, как n:m, n:1, 1:n.

Этапы интеграции

Весь процесс интеграции можно разбить на несколько этапов, последовательность которых показана на рисунке ниже.

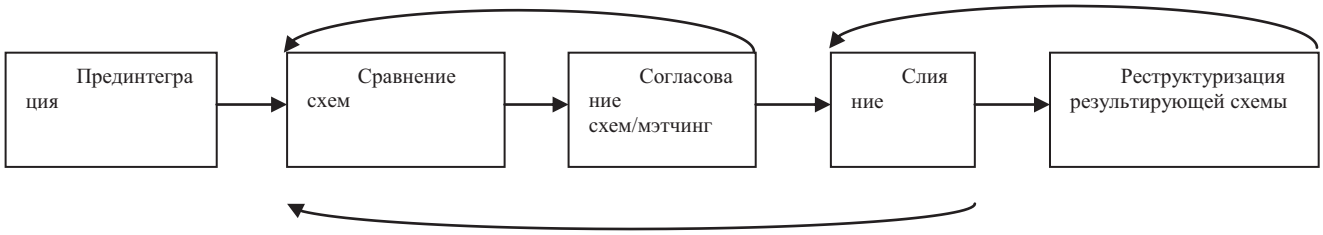

Рис. 3. Этапы интеграции схем данных.

Далее рассмотрим подробно каждый этап.

Прединтеграция

На данном этапе определяется порядок, в котором проводится интеграция схем. На рисунке показаны 4 варианта стратегий, которые описывают порядок, в котором происходит интеграция.

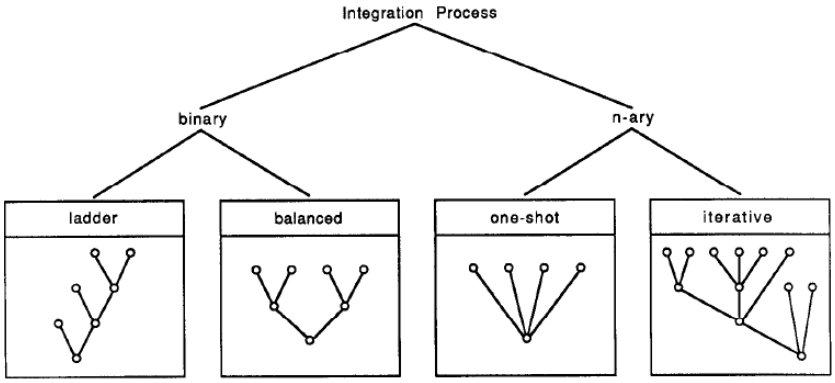

Рис. 4. Стратегии прединтеграции

Стратегии представлены в виде деревьев, в которых листья обозначают входные схемы, не листовые узлы – промежуточные схемы, а корневые узлы представляют собой результирующие схемы данных. Стратегии делятся на две группы – бинарные и n-арные.

Бинарные стратегии позволяют проводить интеграцию только двух схем одновременно. В случае, когда новая входная схема интегрируется с промежуточной, такая бинарная стратегия называется *лестничной (ladder)*. В случае, когда схемы разбиваются на пары и интегрируются симметрично, то такая бинарная стратегия называется *сбалансированной (balanced)*.

В свою очередь n-арные стратегии позволяют интегрировать n схем одновременно, при этом n-арная стратегия *реализуется в один проход (one-shot strategy)* в случае, когда все схемы интегрируются одновременно за один проход. В любом другом случае n-арная стратегия называется *итеративной (iterative)*.

Преимущество бинарных стратегий заключается в простоте их реализации, но в то же время с точки зрения вычислительной сложности алгоритмов этот вариант менее предпочтителен по сравнению с n-арными стратегиями, так предполагает большее число операций сравнения. Преимущество лестничной стратегии над сбалансированной в том, что она позволяет выбирать схемы по определенному порядку, на основе приоритета или важности. Очевидно, что при интеграции смысл каких-то сущностей будет теряться и «удельный вес» и семантика первых отобранных схем в результирующей схеме будет выше.

Преимущество n-арных однопроходных стратегий в их скорости, а так же в том, что семантический анализ можно провести единожды в самом начале – тем самым избегаются конфликты промежуточных этапов лестничной стратегии. Итеративные n-арные стратегии имеет смысл использовать, когда удается разбить все исходные схемы на классы эквивалентности.

Сравнение схем

Основной задачей данного этапа является определение всех конфликтов в представлениях одних и тех объектов в разных схемах. Можно выделить два основных типа конфликтов:

- Конфликты названий (синонимы, омонимы);
- Структурные конфликты.

Для разрешения конфликтов имен используются словари данных, классификаторы и на их основе определяется степень близости атрибутов/сущностей с одинаковыми описаниями. Значительно упрощает ситуацию использование описаний полей – описания могут храниться в таблицах с мета-информацией базы данных.

Структурные конфликты можно разделить на следующие группы:

Конфликты типов. Здесь речь идет не только о различных типах данных одного языка, например NVARCHAR(10) и NVARCHAR(20). Проблемы могут быть намного глубже, когда в одной схеме данных объект моделируемой предметной области представлен сущностью, а в другой схеме данных – атрибутом сущности.

Конфликты зависимостей. Данный тип конфликтов возникает в случае, если не совпадают мощности эквивалентных отношений в интегрируемых схемах. Например, отношение «Живут в браке» между сущностями Мужчина и Женщина в одной схеме может иметь мощность 1:1, а в другой – m:n.

Конфликты ключей. В одной схеме данных ключ может состоять из одного атрибута, а в другой – из нескольких атрибутов. Не ключевой атрибут может семантически соответствовать ключевому атрибуту интегрируемой схемы.

Поведенческие конфликты. Возникают в случае различных политик вставки/удаления соответствующих между собой объектов схемы. Например, в одной схеме данных удаление последнего сотрудника департамента повлечет собой удаление самого департамента, так как не допускается существование департаментов без сотрудников. А в другой схеме данного правила может не быть.

Согласование схем

Цель данного этапа – подготовить схемы к слиянию, то есть разрешить конфликты. Задача разрешения конфликтов, по существу, сводится к определению семантически

соответствующих сущностей/атрибутов и последующего преобразования схем. В случае, если конфликт не может быть разрешен, то об этом отправляется уведомление пользователю, который вручную разрешает его. Преобразование схем можно осуществлять за счет выполнения следующих операций:

- Удаление избыточных ограничений целостности;
- Удаление избыточных отношений между сущностями;
- Агрегирование;
- Детализация;
- Создание дополнительных обобщающих сущностей (генерализация);
- Создание супертипов;
- Создание подтипов;
- Инкапсулирование;
- Создание дополнительных атрибутов;
- Удаление функций/триггеров.

На рисунке ниже показаны два случая преобразования атрибута в сущность:

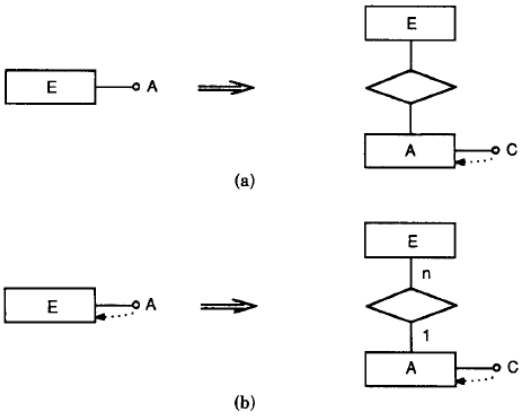

Рис. 5. Преобразование атрибута в сущность

Пунктирная линия на рисунке 5b указывает на то, что атрибут A является ключевым, в этом случае при формировании сущности A мощность отношения между A и E определена как 1:n, что характерно для всех ключевых атрибутов.

Таким образом, для каждого типа конфликтов необходимо разработать методику разрешения. Фундаментальной проблемой в этом случае является то, что среди всех существующих методик нет ни одной, в которой была бы доказана функциональная полнота преобразований. Большинство известных подходов реализованы на базе эвристических алгоритмов, а не на базе строгих математических правил.

Слияние и реструктуризация

На данном этапе происходит создание промежуточной результирующей схемы данных (после этапа слияния) и ее преобразование, т.е. реструктуризация для того, чтобы она удовлетворяла определенным технологическим и пользовательским характеристикам. Слияние согласованных схем можно представить как их наложение одна на другую, когда совпадающие элементы не дублируются, а отражаются в один единственный.

Целью слияния и реструктуризации является обеспечение следующих характеристик:

Минимальность. Задача достижения минимальность сводится к поиску и устранению избыточностей.

На рисунке ниже показана схема данных с избыточной связью.

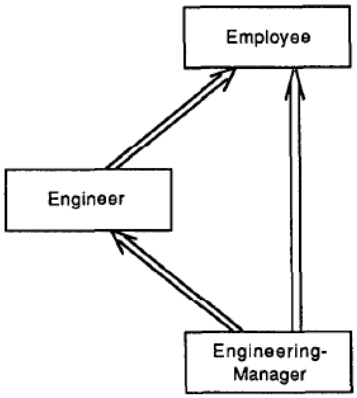

Рис. 6. Схема данных с избыточной связью

Из рисунка видно, что связь между сущностями Engineering-Manager и Employee избыточна, так как она может быть наследована из связи Engineering-Manager и Engineer.

Понятность. Обеспечение данного свойства лучше показать на примере, показанном на рисунке ниже.

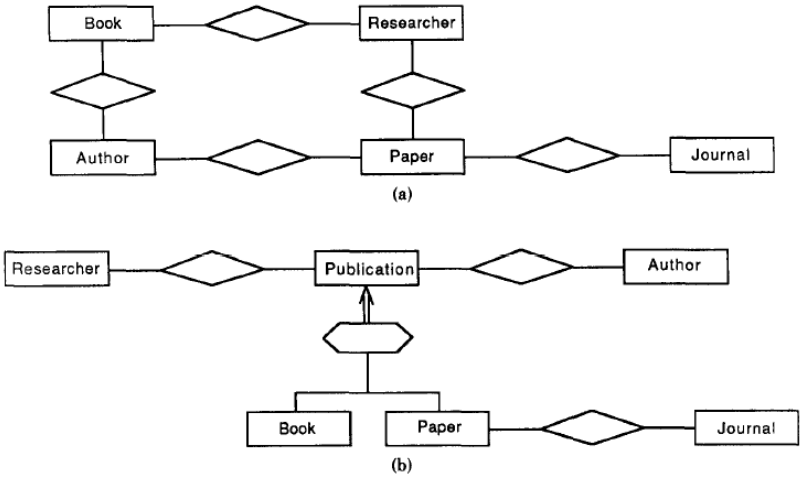

Рис. 7. Улучшение понятности: исходные данные (a) и результат (b)

Понятность схемы была улучшена за счет введения дополнительной сущности-супертипа Publication, а сущности Book и Paper наследуются от сущности Publication.

Список литературы

1. Брешенков А.В., Балдин А.В. Анализ проблемы проектирования реляционных баз данных на основе использования информации табличного вида и разработка модели методики проектирования. М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. -150 с.
2. Брешенков А.В. Методы решения задач проектирования реляционных баз данных на основе использования существующей информации табличного вида. М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. -150 с.
3. From Databases to Dataspaces: A New Abstraction for Information Management Michael Franklin University of California, Berkeley Alon Halevy, Google Inc.
4. AL-FEDAGHI, S., AND SCHEUERMANN, P. 1981. Mapping considerations in the design of schemas for the relational model. IEEE Trans. Softw. Eng. SE-7, 1 (Jan.).
5. BATINI, C., AND LENZERINI, M. 1984. A methodologyfor data schema integration in the entity relationship model. IEEE Trans. Softw. Eng. SE-lo, 6 (Nov.), 650-663.
6. A Comparative Analysis of Methodologies for Database Schema Integration, C. Batini, M. Lenzerini and S.B. Navathe, ACM Computing Surveys, 18, 4, December 1986, pp. 323-364
7. A Survey of Approaches to Automatic Schema Matching, Erhard Rahm, Philip A. Bernstein,The International Journal on Very Large Data Bases, 10(4):334-350.