

Объектные расширения SQL «объектны»!

Сергей Кузнецов, ИСП РАН

10 лет спустя (1)

- ▶ 23 декабря 2004 г.
- ▶ **Объектно-ориентированные базы данных и объектные расширения языка SQL**
- ▶ Более подробно: "Объектны" ли объектные расширения языка SQL?, 2005
 - ▶ http://citforum.ru/database/articles/sql_odmg/
- ▶ «Единственное, что объединяет модели данных SQL и ODMG, -- развитая система типов с возможностью определения новых типов с произвольно сложной структурой значений. "Объектные" же расширения SQL, по сути, представляют собой дальнейшую синтаксическую маскировку операций естественного соединения.»

10 лет спустя (2)

- ▶ За 10 лет представления изменились
- ▶ Многие размышления привели к следующим выводам:
- ▶ Различия, которые мне казались глубинными, таковыми не являются, да и вообще не являются различиями уровня модели
- ▶ Объектные расширения SQL обеспечивают не меньшие (а скорее большие) возможности, чем объектная модель ODMG
- ▶ При разумном (с позиций сообщества баз данных) использовании СУБД, основанной на модели ODMG, будут создаваться базы данных и средства манипулирования ими, близкие к тем, которые предписывает модель данных SQL

ODMG 3.0 (основные понятия) (1)

- ▶ Две категории типов - литеральные и объектные
- ▶ Литеральные типы обычные - множество значений и набор операций
- ▶ Атомарные литеральные типы (целые, плавающие, строки, булевский тип)
- ▶ Булевский тип двузначный, неопределенных значений в явном виде нет (хотя в скрытом виде встречаются)
- ▶ Структурные литеральные типы (как в Паскале и Си)
- ▶ Литеральные типы коллекций - множества, мультимножества, списки, справочники
- ▶ От типов упорядоченных коллекций открешиваюсь
 - ▶ Не люблю, боюсь и не понимаю смысла

ODMG 3.0 (основные понятия) (2)

- ▶ Атомарные объектные типы и объектные типы коллекций
- ▶ Атомарный объектный тип:
 - ▶ атрибуты
 - ▶ связи
 - ▶ операции
- ▶ Отличается от традиционного типа тем, что непонятно множество значений
- ▶ Большой дефект модели ODMG - отсутствие ссылочных типов
 - ▶ одно дело - объект, созданный операцией new, другое - значение атрибута объектного типа или элемент коллекции
 - ▶ о последнем случае неявно имеется в виду OID
 - ▶ зачем нужны двунаправленные связи (OID -однонаправленный «адрес»)

ODMG 3.0 (основные понятия) (3)

- ▶ Экстент атомарного объектного типа
- ▶ Наследование (множественное на уровне спецификаций, одиночное на уровне реализаций)
- ▶ Экстент подтипа является подэкстентом супертипа (из-за семантики включения)
- ▶ Объектные типы коллекций (существенны множества, зачем мультимножества - непонятно)
- ▶ Самое важное «самоприменение» - экстененты

Объектная часть модели SQL (1)

- ▶ Про типы в SQL в целом не говорим
- ▶ Для объектной части важен определяемый пользователем структурный тип
 - ▶ Традиционный тип
 - ▶ Набор типизированных атрибутов (множество значебний)
 - ▶ Набор операций
 - ▶ Определение парного ссылочного типа
- ▶ Наследование (одиночное, похожее на Java)
- ▶ Сам по себе структурный тип не относится только к объектной части
 - ▶ Можно определять столбцы обычных таблиц структурного типа

Объектная часть модели SQL (2)

- ▶ Типизированные таблицы (Create table of type)
 - ▶ С одной стороны, обычная таблица с $n+1$ столбцом
 - ▶ С другой стороны - множество объектов
 - ▶ «самоссылающийся» столбец содержит ссылочные значения (ссылочного типа) - аналог OID
- ▶ Ссылочные значения могут храниться в столбцах других таблиц (атрибутах объектов)
- ▶ Можно переходить по ссылке (выполнять операцию разыменования)
 - ▶ Обычно - естественное соединение
 - ▶ Но в Oracle переход по rowid (прямой адрес)

Объектная часть модели SQL (3)

- ▶ Наследование типизированных таблиц
- ▶ Таблица может быть подтаблицей супертаблицы, если ее тип является непосредственным подтипом типа супертаблицы
- ▶ Зачем нужно явное определение подтаблиц?
- ▶ Может быть несколько типизированных таблиц одного типа (несколько экстенстов)
 - ▶ Более правильно сделано создание объекта
 - ▶ Insert (аналог new) - операция над таблицей, а не операция типа
 - ▶ Когда это может быть полезно?

Разные способы адресации (1)

- ▶ 10 лет назад мне казалось, что принципиальным различием между ODMG и SQL является использование в ODMG прямой адресации, а в SQL - ассоциативной
- ▶ В ODMG OID хранится только на стороне ссылающейся сущности
 - ▶ Разыменование происходит по фон Нейману
 - ▶ Собственно, поэтому нужно отдельное понятие двунаправленной связи
- ▶ В SQL ссылочное значение хранится в ссылающейся сущности и в объекте
 - ▶ Разыменование происходит на основе естественного соединения (или фильтрации - частный случай)
 - ▶ Поэтому отдельное понятие связи не требуется

Разные способы адресации (2)

- ▶ Несколько лет назад я понял, что это не важно
- ▶ 40 лет в фоннеймановских компьютерах имеется кэш
- ▶ Поиск в нем происходит ассоциативно (и параллельно)
- ▶ Из-за этого компьютеры не перестают считать фоннеймановскими
- ▶ Можно на уровне реализации ODMG сделать разыменовывание ассоциативным (и хранить такие OID в объекте)
- ▶ Можно на уровне реализации SQL сделать разыменовывание прямым (можно хранить ссылочное значение в объекте, можно не хранить - как в Oracle)
 - ▶ Правда, в этом случае неясно, как быть со связями
- ▶ Так что различие в адресации не является модельным различием

Что хранится в экстенте (или в типизированной таблице)?

- ▶ 10 лет назад мне казалось, что модельным различие между ODMG и SQL является то, что в ODMG каждый объект получает место из общей «кучи», а в SQL новый объект получает место в одной из существующих типизированных таблиц
- ▶ Но на самом деле для объектного мира важно свойство object identity
- ▶ В ODMG можно поддерживать отдельную кучу для каждого экстента, и хранить в экстентах (вплоть до физического уровня) объекты, а не их OID
- ▶ В SQL можно (и так и делается на физическом уровне) распределять память для строк почти что в общей куче, а таблицу целиком собирать на логическом уровне
- ▶ Так что и различие с способе распределения памяти под объекты модельным не является
- ▶ И больше модельных различий я не вижу

СУБД, соответствующая стандарту SQL, является (в частности) объектно-ориентированной

- ▶ В SQL есть все, что требуется канонами объектно-ориентированного мира
- ▶ Печально, что скоро придется писать статью «Почему уже 20 лет не используются объектные возможности SQL?»
- ▶ Странные люди продолжают пользоваться ORM+SQL:1992, а не объектно-ориентированными СУБД или SQL:2003
- ▶ Не совсем понятно, как писать клиентские объектно-ориентированные приложения, но ясно, что серверные приложения писать можно вполне удобно
- ▶ Наверное, стоит прислушаться к тому, что говорил Джим Грей: подождите, люди ленивы, инертны и неинициативны - рано или поздно они поумнеют
- ▶ Хотелось бы дожить до этого

ODMG (при разумном использовании) – это упрощенный SQL

- ▶ Сторонники ООСУБД говорят, что в SQL в базе данных лежат плоские структуры (таблицы), а в ODMG – объекты разного типа и сложности
- ▶ Но все объекты в БД ODMG (кроме экстентов) безымянны
 - ▶ Как работать с такой базой данных?
- ▶ Кроме того, OQL выродится в Xpath
- ▶ Для ненавигационности нужна массивная регулярность
- ▶ Так что при разумном использовании ODMG БД – это набор именованных экстентов
 - ▶ Читай: типизированных таблиц
- ▶ Но SQL может намного больше

В ближайшие 10 лет постараюсь об этом не
говорить

С НОВЫМ ГОДОМ!

