



Начальной стадией проектирования системы баз данных является построение семантической модели предметной области, которая базируется на анализе свойств и природы объектов предметной области и информационных потребностей будущих пользователей разрабатываемой системы. Эту стадию принято называть концептуальным проектированием системы, а ее результат – концептуальной моделью предметной области (объектом моделирования здесь является предметная область будущей системы).

Такие модели обобщенно представляют информационные потребности пользователей создаваемой системы в части использования хранимых данных и по существу являются средством коммуникации как разработчиков, так и пользователей на разных стадиях жизненного цикла базы данных.

Назначение концептуальных моделей определяет и некоторые специфические требования к средствам их представления. Помимо упомянутой независимости от среды (оборудования) и требования адекватности отражения предметной области отметим следующие:

- формализованность, обеспечивающую возможность автоматизированной обработки, в том числе, например, автоматический контроль непротиворечивости;
- дружелюбность, обеспечивающую возможность использования наглядных графических средств отображения и обработки их пользователем.

Проблема проектирования базы данных концептуальная модель

К концептуальным моделям относятся различные компоненты, по-разному и разными средствами отражающие предметную область. Помимо наиболее известного описания объектов и связей между ними (модель «сущность-связь») к концептуальному уровню описания предметной области можно отнести следующие компоненты:

- систему атрибутов и средств описания предметной области. Например, логические (автоматические) связи между показателями или лингвистические свойства языка (синонимия, синтаксис и т.д.), используемую для вербального представления объектов;
- ограничения целостности, определяющие допустимость значения отдельных полей и взаимосвязей как на уровне семантики содержимого БД, так и ее физической структуры (отдельных файлов данных и взаимосвязей между ними);

- описание информационных потребностей пользователей, например, в виде типовых запросов, отражающих процедурные особенности обращения к данным.

Одной из наиболее популярных средств формализованного представления предметной области систем, ориентированных на обработку фактографической информации, является модель «сущность-связь», которая положена в основу значительного количества коммерческих CASE-продуктов, поддерживающих полный цикл разработки систем баз данных или отдельные его стадии. При этом многие из них не только поддерживают стадию концептуального проектирования предметной области разрабатываемой системы, но и позволяют осуществить на основе построенной их средствами модели стадию логического проектирования путем автоматической генерации концептуальной схемы базы данных для выбранной СУБД, например, схемы базы данных для какого-либо SQL-севера или объектной СУБД.

Моделирование предметной области в этом случае базируется на использовании графических диаграмм, включающих сравнительно небольшое число компонентов, и самое важное – технологию построения таких диаграмм.

Семантическую основу ER-модели составляют следующие предположения:

- та часть реального мира (совокупность взаимосвязанных объектов), сведения о которых должны быть помещены в базу данных, может быть представлена как совокупность сущностей;
- сущности можно классифицировать по типам сущностей: каждый экземпляр сущности (представляющий некоторый объект) может быть отнесен к классу – типу сущностей, каждый экземпляр которого обладает общими для них и отличающими их от сущностей других классов свойствами;
- систематизация представления, основанная на классах, в общем случае предполагает иерархическую зависимость типов: сущность А является подтипом сущности В, если каждый экземпляр типа А является экземпляром сущности типа В;
- взаимосвязи объектов могут быть представлены как связи – сущности, которые служат для фиксирования (представления) взаимосвязанности двух или нескольких сущностей.

Здесь следует еще раз подчеркнуть информационную природу понятия сущность и его соотношение с материальными или воображаемыми объектами предметной области. Любой объект предметной области обладает свойствами, часть из которых выделяется как характеристические – значимые с точки зрения

прикладной задачи. При этом, например, в процессе анализа и систематизации предметной области обычно выделяются классы – совокупность объектов, обладающих одинаковым набором свойств, задаваемых в виде наборов атрибутов (значения атрибутов для объектов одного класса, естественно, могут различаться). Соответственно, на уровне представления предметной области (т.е. ее концептуальной модели) объекту, рассматриваемому как понятие (объект в сознании человека), соответствует понятие сущность; объекту как части материального мира (и существующему независимо от сознания человека) соответствует понятие экземпляр сущности; классу объектов соответствует понятие тип сущности.

В дальнейшем, поскольку в концептуальной модели рассматриваются не отдельные экземпляры объектов, а классы, мы не будем различать соответствующие понятия этих двух уровней, т.е. будем предполагать тождественность понятий объект и сущность, свойство объекта и свойство сущности.

ER-модель, как описание предметной области, должна определить объекты и взаимосвязи между ними, т.е. установить связи следующих двух типов.

1. Связи между объектами и наборами характеристических свойств, и таким образом определить сами объекты.
2. Связи между объектами, задающие характер и функциональную природу их взаимосвязанности.

Наиболее часто на практике семантическое моделирование используется на первой стадии проектирования базы данных. При этом в терминах семантической модели производится концептуальная схема базы данных, которая затем вручную преобразуется к реляционной (или какой-либо другой) схеме. Этот процесс выполняется под управлением методик, в которых достаточно четко оговорены все этапы такого преобразования.

Менее часто реализуется автоматизированная компиляция концептуальной схемы в реляционную. При этом известны два подхода: на основе явного представления концептуальной схемы как исходной информации для компилятора и построения интегрированных систем проектирования с автоматизированным созданием концептуальной схемы на основе интервью с экспертами предметной области. И в том, и в другом случае в результате производится реляционная схема базы данных в третьей нормальной форме (более точно следовало бы сказать, что автору неизвестны системы, обеспечивающие более высокий уровень нормализации).

Наконец, третья возможность, которая еще не вышла (или только выходит) за пределы исследовательских и экспериментальных проектов, - это работа с базой данных в семантической модели, т.е. СУБД, основанные на семантических моделях данных. При этом снова рассматриваются два варианта: обеспечение пользовательского интерфейса на основе семантической модели данных с автоматическим отображением конструкций в реляционную модель данных (это задача примерно такого же уровня сложности, как автоматическая компиляция концептуальной схемы базы данных в реляционную схему) и прямая реализация СУБД, основанная на какой-либо семантической модели данных. Наиболее близко ко второму подходу находятся современные объектно-ориентированные СУБД, модели данных которых по многим параметрам близки к семантическим моделям (хотя в некоторых аспектах они более мощны, а в некоторых - более слабы).

Средства автоматизированного проектирования концептуальной модели привлекают к себе в настоящее время большой интерес и используются в процессе создания структуры базы данных и интерфейса пользователя для доступа к данным.

Причина применения этих средств состоит в использовании в подавляющем большинстве реальных разработок баз данных спиральной модели жизненного цикла программного обеспечения, что предусматривает последовательное создание нескольких версий программного обеспечения. Каждая следующая версия включает в себя предыдущую (возможно, не полностью) и является ответом на замечания пользователя, полученные в результате тестирования предыдущей версии. При создании баз данных первая модель программного обеспечения, к сожалению, очень редко является удачной. Чаще всего заказчик отвергает первую версию, так как она недостаточно полно отвечает его требованиям. Причина такой ситуации заключается в том, что заказчик не может сразу, до создания начальной версии программы, четко и полно сформулировать свои требования. Обычно после получения первого варианта программного обеспечения заказчик выдвигает дополнительные требования, которые нельзя реализовать в рамках созданной базы данных. Это вынуждает разработчиков вносить изменения в структуру базы данных, а также, соответственно, в интерфейс пользователя для доступа к базе данных. Таких итераций может быть несколько до момента получения решения, адекватного запросам заказчика. Но даже после получения удовлетворительного решения процесс разработки базы данных не завершается. Жизнь не стоит на месте, и запросы заказчика меняются с течением времени. Часть этих изменений можно реализовать без изменения структуры базы данных, изменяя только

интерфейс пользователя, другие же требуют изменения и интерфейса, и структуры базы данных. Надо заметить, что подобные изменения являются очень болезненными – работа по их внесению может оказаться трудоемкой и, что самое неприятное, потребовать замены большого количества отлаженного программного кода. Иными словами, замененный код был написан впустую, на самом деле его не нужно было писать.

Таким образом, создание работоспособной базы данных можно условно разделить на три этапа – проектирование базы данных, в процессе которого создаются рабочие прототипы, кодирование – создание структур баз данных и законченного интерфейса пользователя и сопровождение готовой базы данных.

Заключение

Основная идея применения средств автоматизированного проектирования баз данных заключается в том, что процесс ручного кодирования начинается только после окончания процесса проектирования. На стадии проектирования схема базы данных и интерфейс пользователя для доступа к базе данных создаются автоматически, исходя из описания концептуальной модели, с помощью так называемых CASE-средств (Computer Aided Software/System Engineering). Конечно, созданный таким образом интерфейс не является законченным программным продуктом, однако он позволяет заказчику оценить возможности конечного продукта и внести свои коррективы. Только после одобрения заказчиком рабочего прототипа разработчики приступают к ручному кодированию – созданию законченного приложения.

При сопровождении все повторяется, за тем исключением, что генерируется не все приложение целиком, а только часть, которую надо изменять.

На практике чаще всего CASE-средства используются для создания схемы базы данных в виде ER-диаграмм и генерации структур баз данных для конкретной СУБД. После получения от заказчика изменений разработчики вносят соответствующие исправления в диаграмму "сущность – связь" и заново генерируют структуры баз данных. Средства автоматической генерации интерфейсов используются реже.

Кроме того, на рынке представлены решения третьих фирм, не производящих СУБД. Одними из самых распространенных являются программные продукты

фирмы AllFusion – AllFusion Erwin Data Modeler и AllFusion Process Modeler (ранее – BPwin) и другие. На российском рынке данные программы предлагает фирма Interface Ltd. Создание диаграммы "сущность –связь" осуществляется с помощью AllFusion Erwin Data Modeler, дальнейшее моделирование, включая генерацию программного кода создания базы данных производится с помощью программы AllFusion Process Modeler.

Создав наглядную модель базы данных можно оптимизировать структуру БД и добиться её полного соответствия требованиям и задачам организации.

Визуальное моделирование повышает качество создаваемой базы данных, продуктивность и скорость её разработки.

Список литературы:

1. http://citforum.ru/cfin/prcorpsys/infsistpr_09.shtml
2. <https://www.intuit.ru/studies/courses/508/364/lecture/8649?page=4>