

Владислав Юрьевич Пирогов
г. Шадринск

Некоторые вопросы преподавания основ реляционных баз данных

В статье рассматриваются некоторые общие принципы преподавания основ реляционных баз данных. Автор подчеркивает важность связи процесса проектирования баз данных с требованиями к информационной системе, неотъемлемой частью которой является проектируемая база данных. Рассматриваются также вопросы преподавания некоторых разделов реляционного анализа, в частности тема «Нормальные формы». Автор отмечает, что для полного понимания требований нормальных форм необходимо обращаться к анализу предметной области, а не ограничиваться только формальным рассмотрением. В статье обсуждаются также некоторые аспекты сильных и слабых сущностей в модели «сущность-связь». Разбираются некоторые подходы, позволяющие максимально приблизить изучение реляционных баз данных к практическому применению в разработке информационных систем. В статье описываются только некоторые вопросы изучения реляционных баз данных, которые кажутся автору важными.

Ключевые слова: реляционные базы данных, СУБД, нормальные формы, проектирование баз данных, преподавание баз данных, ER-модель.

Vladislav Yurievich Pirogov
Shadrinsk

Aspects of teaching the basics of relational databases

The article views some general principles of teaching the basics of relational databases. The author emphasizes the importance of linking the database design process with the requirements for the information system of which the database being designed is an integral part. The author discusses the issues of teaching some sections of relational analysis in particular the topic of "Normal forms". The author notes that in order to fully understand the requirements of normal forms, it is necessary to turn to the analysis of the subject area, and not be limited only to formal consideration. The article also stresses some aspects of strong and weak entities in the entity-relationship model. Some approaches are being analyzed that make it possible to bring the study of relational databases as close as possible to practical application in the development of information systems. The author describes only some important aspects of studying relational databases.

Keywords: relational databases, DBMS, normal forms, database design, database teaching, ER-model.

Современные информационные системы (далее ИС), строятся на использовании средств хранения больших объемов информации и средств оперативного доступа к хранимым данным [1]. Основным инструментом работы с хранилищами данных являются системы управления базами данных (СУБД), чаще всего поддерживающие реляционную модель данных. Основы реляционных баз данных (РБД) [1-8] в той или иной степени излагаются в широком спектре дисциплин, преподаваемых на инженерных специальностях, связанных с разработкой программного обеспечения: Информационные системы, Разработка информационных систем, Базы данных, WEB-программирование и др. Сложность преподавания раздела, касающегося реляционных баз данных, заключается прежде всего в том, что это требует большой практической работы. С одной стороны необходима выработка компетенций проектирования баз данных на основе анализа предметной области, а с другой стороны здесь традиционно принято излагать материал на основе формальных математических положений. Традиционные разделы практически любой книги по РБД обычно содержат такие разделы как Реляционная алгебра и Реляционные исчисления, а также известную тему «Нормальные формы» [1, 2, 3]. Мы рассмотрим некоторые подходы, позволяющие максимально приблизить изучение реляционных баз данных к практическому

применению в разработке информационных систем. В краткой статье мы описываем только некоторые вопросы изучения РБД, которые кажутся нам важными.

Общие подходы

При изложении вопросов, связанных с проектированием баз данных, следует придерживаться важного методологического принципа. Кратко это можно сформулировать как анализ предметной области на основе постановки задачи. Последнее тесно связано с тем, какую информационную систему (ИС) мы разрабатываем, а это в конечном итоге сводится к набору требований к ИС. На этот принцип следует постоянно обращать внимание студентов, указывая на то, как он в значительной степени определяет структуру базы данных.

При анализе предметной области принято использовать парадигму, основанную на модели «сущность-связь» (ER-модель) [1, 2]. Элементами этой модели являются: сущность, атрибуты сущности, связи между сущностями. Этот простой набор понятий позволяет глубоко анализировать предметную область, а затем легко переходить уже к реляционной базе данных, которая будет соответствовать постановке задачи.

В качестве примера того, как важна предварительная постановка задачи, студентам предлагается проанализировать предметную область: учебный процесс — студенты, преподаватели, предметы. При этом не

конкретизируются требования к ИС, частью которой должна стать проектируемая база данных. Обычно студенты рисуют связь между студентом и преподавателем, но почти всегда затрудняются ответить на вопросы: А что это за связь? Чем она обусловлена? В лучшем случае студент пытается нарисовать связь, как связь между сильной и слабой сущностями. Сильной является сущность Преподаватель, а слабой сущность Студент. С обывательской точки зрения это так и есть. Но следует обратить внимание, что пока нет конкретных требований к решаемой задаче (к будущей ИС), трудно понять, как связаны эти сущности и связаны ли они вообще. Предлагается три варианта конкретизации задачи:

1. Задача предполагает учёт оценок студента. Следовательно связь обусловлена выставлением оценок преподавателем студенту. Сама связь, таким образом, есть лишь отражением документов: зачётная книжка и ведомость оценок.

2. Задача предполагает учёт расписания занятий. И тогда связующим элементом между двумя сущностями (и это опять документ) является расписание. При этом появляется такая дополнительная сущность, как учебная группа.

3. В задаче предполагается учитывать учебный план, по которому занимаются студенты конкретной специальности и учебную нагрузку преподавателей. Здесь более сложные взаимоотношения между рассматриваемыми сущностями, но опять связь вполне конкретна и связана с документами.

Далее показывается, что рассматриваемая связь – это связь «многие ко многим», которая сама является сущностью. Более того, как было показано выше, она воплощает в себе вполне конкретный объект(ы) предметной области.

Рассматривая проектирование баз данных, следует указать последовательность проектирования:

1. Выделение сущностей в предметной области.

2. Определение атрибутов сущностей.

3. Анализ атрибутов на предмет разбиение их на несколько атрибутов.

4. Анализ атрибутов с точки зрения возможного превращения их в сущности.

5. Определение и анализ связей между сущностями, что, как правило, приводит к появлению новых сущностей.

6. Преобразование ER-модели к реляционной базе данных.

О преподавании темы Нормальные формы

Раздел реляционных баз данных, называемый «Нормальные формы», посвящён формальным требованиям к таблицам, позволяющим производить их декомпозицию, в случае невыполнения требований [1, 3]. Подход формальный и студентам не всегда ясен смысл этих требований и их практическое применение. Мы предлагаем студентам, прежде чем

анализировать таблицу на предмет выполнения требований нормальных форм, провести процесс обратный процессу проектирования. т.е. понять на основе какой предметной области и какой постановки задачи могла получиться такая таблица. После чего снова провести процесс проектирования. Такой подход понятен и естественен с практической точки зрения и приводит:

1. К получению реляционной базы данных, в таблицах которой отсутствует нарушение требований нормальных форм.

2. К более глубокому пониманию самой темы «Нормальные формы».

Требования 1-й нормальной формы сводятся к необходимости того, чтобы на пересечении строки столбца таблицы находился неделимый (атомарный) элемент [1]. Чтобы понять это требование, нужно вернуться к общему положению, которое гласит, что анализ следует проводить на основе постановки задачи. Поэтому неделимость элемента определяется только постановкой задачи. Если в постановке задачи не требуется оперировать отдельно фамилией, именем и отчеством, то элемент <ФИО>, т.е. строка содержащая фамилию, имя и отчество человека, в данном контексте можно считать неделимым. Следовательно, нарушается или нет первая нормальная форма в таблице определяется предметной областью и постановкой задачи, а не абстрактными представлениями. При этом студент должен понимать, что для того, чтобы сделать такой вывод пришлось реконструировать предметную область и определиться с тем, какие требования мы предъявляем к тому, как будет использована таблица (требования к информационной системе).

Требования 2-й нормальной формы. Находящаяся в первой нормальной форме таблица будет находиться во второй нормальной форме, если любой ее атрибут, не входящий в состав первичного ключа, функционально полно зависит от этого первичного ключа [1, 2]. *Понятие функциональной зависимости является фундаментальным понятием теории реляционных баз данных. Определение функциональной зависимости можно найти в [1, 2, 3].*

Конечно, здесь появляется термин «функционально полно», но однако даже не вдаваясь в суть этого понятия, очевидно, что это требование выполняется, если выбранный первичный ключ является простым и суррогатным. *Суррогатный ключ — ключ не зависящий от предметной области [1]. Очень часто (но не всегда) в качестве суррогатного выбирается счётчик добавляемых строк.* В нашей практике с самого начала студент знает это требование. Т.о. требование второй нормальной формы выполняется автоматически. И приведение таблицы к требованиям второй

нормальной формы, т.о. не составляет никакого труда.

Требования 3-й нормальной формы.

Таблица, находящаяся во второй нормальной форме, будет находиться в третьей нормальной форме, если все ее атрибуты, не входящие в первичный ключ, не зависят транзитивно от первичного ключа [1, 2].

Типичный пример таблицы с транзитивной зависимостью
 <Номер_поставки, Номер_поставщика, Город_поставщика, Дата_поставки>. Здесь атрибут город_поставщика транзитивно зависит от атрибута номер_поставки. Но даже не вникая в суть транзитивной зависимости путём простого анализа предметной области, легко увидеть, что в одной сущности содержатся три: <Номер_поставки, Дата_поставки>, <Номер_поставщика>, <Город>. После этого задача сводится к определению связей между этими сущностями. Подчеркнём ещё раз, что к

такой декомпозиции мы приходим не на основе формальных требований нормальной формы, а путём содержательного анализа предметной области.

Требование 4-й нормальной формы.

Пропускаем известную в реляционном анализе нормальную форму Бойса-Кодда [8], т.к. её рассмотрение не даёт принципиального дополнения нашему изложению. Эта нормальная форма связана с понятием множественной зависимости, которая является обобщением функциональной зависимости [1]. Рассмотрим пример. <Поставщик, Заказчик, Магазин>, магазины и поставщики зависят от конкретного заказчика. Как и в предыдущем случае решить проблему можно выделив сущности данной предметной области. После этого задача сводится к правильному указанию связей между этими сущностями и в конечном итоге таблицами.



Рис 1: Пример схемы данных для 4-й нормальной формы

На рисунке 1 представлена схема РБД, демонстрирующая рассмотренный выше пример. Показано разрешение проблемы с 4-й нормальной формой с учётом анализа предметной области.

Требования 5-й нормальной формы. Типичным примером нарушения 5-й нормальной формой является таблица <Отдел, Сотрудник, Проект>. Предполагается, что 1. Сотрудник может

одновременно числиться в разных отделах. 2. Сотрудник, как и отдел могут являться исполнителями проектов. Интересно, что здесь нет ничего нового с точки зрения проектирования базы данных. И если взять за основу предметной области сущности: Отдел, Сотрудник, Проект то задача сводится к определению связей между ними (см. Рисунок 2).

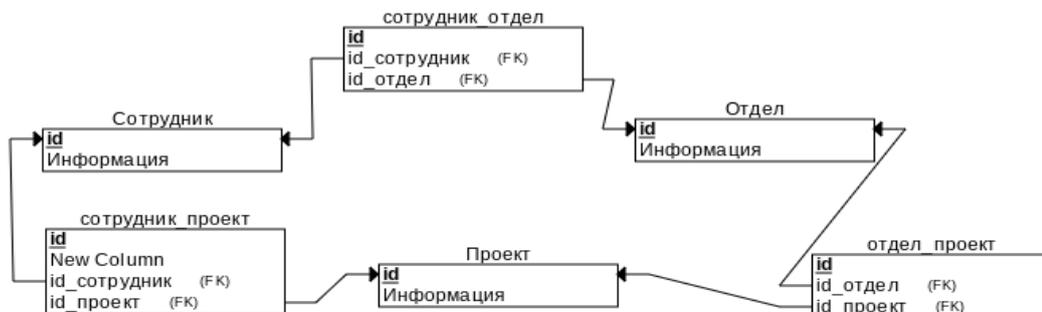


Рис.2. Пример устранения проблемы нарушения 5-й нормальной формы

Об особенностях ER-модели и первичные ключи

Говоря о модели «сущность-связь», позволяющей графически отобразить результат анализа предметной области, следует обратить внимание студентов на два важных аспекта:

1. Те результаты анализа предметной области, которые мы отображаем ER-диаграммой, те смыслы которые она содержит, при переходе к реляционной базе данных, как правило, частично теряются. И, таким образом, ER-модель не только удобное средство анализа предметной области, но и возможность сохранить результаты этого анализа для дальнейшей модернизации системы.

2. Понятие сильной и слабой сущности всегда относительно. Слабая сущность по отношению к одной сущности, может оказаться сильной по отношению к другой [1].

На последнем положении следует остановиться особо. Дело в том, что понятие сильной и слабой сущности как раз и является результатом анализа предметной области. При переходе к реляционному (табличному) представлению признак «силы» и «слабости» становятся формальными. Мы обычно говорим, что сильная сущность будет представлена таблицей, первичный ключ которой участвует в реализации связи с другой таблицей. Последняя считается слабой, так как у ней в реализации связи участвует внешний ключ. И вот здесь в некоторых источниках можно прочесть, что таблица, представляющая слабую сущность, не имеет (или не должна иметь) первичного ключа. Методологически это ошибочное утверждение. Поскольку по отношению к другим таблицам она может представлять сильную сущность. Примером может служить соотношение трех

сущностей: Студент, Оценка, Факультет. По отношению к таблице Оценка таблица Студент представляет сильную сущность, тогда как по отношению к таблице Факультет слабую. Единственно верным подходом, на который мы всегда акцентируем внимание студентов, является то, что каждая таблица в РБД всегда должна иметь первичный ключ. Отсутствие первичного ключа в таблице также противоречит известному положению РБД [1, 2], по которому в таблице не может быть двух одинаковых строк.

Заключение

В статье были рассмотрены некоторые аспекты преподавания основ реляционных баз данных для студентов инженерных специальностей. Отметим некоторые важные выводы.

1. Проектирование баз данных должно быть тесно увязано с разработкой информационной системы, одной из важнейших частей которой является создаваемая база данных.

2. Любое проектирование реляционных баз данных следует основывать на анализе предметной области, в основе которого лежат требования к создаваемой информационной системе.

3. В концепцию изложения реляционного анализа и, в частности, теории нормальных форм, следует в первую очередь положить проектирование на базе анализа предметной области, т.е. проектирования сверху вниз.

4. Важность ER-модели обусловлена тем, что при переходе к реляционной базе данных утрачиваются некоторые смыслы, которые эта модель содержит.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Пирогов, В.Ю. Информационные системы и базы данных / В.Ю. Пирогов. – Санкт-Петербург: БХВ, 2009. – 528 с. – Текст : непосредственный.
2. Дейт, К.Дж. Введение в системы баз данных / К.Дж. Дейт. – 8-е изд. – Москва : Вильямс, 2005. – 1328 с. – Текст : непосредственный.
3. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К.Бегг. – 3-е изд. – Москва : Вильямс, 2003. – 1436 с. – Текст : непосредственный.
4. Мирошниченко, Г. Реляционные базы данных: практические приемы оптимальных решений / Г. Мирошниченко. – Санкт-Петербург : БХВ, 2005. – 392 с. – Текст : непосредственный.
5. Стружкин, Н. Базы данных. Проектирование : учебник / Н. Стружкин, В.Годин. – Москва : Юрайт, 2017. – 478 с. – Текст : непосредственный.
6. Codd, E.F. A Relational Model of Data for Large Shared Data Banks / E.F. Codd // IBM Research Laboratory. – San Jose, California : February Communications of the ACM (CACM), 1970.
7. Codd, E.F. Relational. Completeness of Data Base Sublanguages / E.F. Codd // Data Base Systems, Courant Computer Science Symposia Series 6. – Englewood Cliffs, N.S. : Prentice-Hall, 1972.
8. Codd, E.F. The relation model for Database Model / E.F. Codd. – Addison-wesley Publishing Company, 1990. – 540 p.

REFERENCES

1. Pirogov V.Ju. Informacionnye sistemy i bazy dannyh [Information systems and databases]. Saint Petersburg: BHV, 2009. 528 p.
2. Dejt K.Dzh. Vvedenie v sistemy baz dannyh [Introduction to database systems]. Mjscow: Vil'jams, 2005. 1328 p.

3. Konnolli T., Begg K. Bazy dannyh. Proektirovanie, realizacija i soprovozhdenie. Teorija i praktika [Databases. Design, implementation and maintenance. Theory and practice]. Mjscow: Vil'jams, 2003. 1436 p.
4. Miroshnichenko G. Reljacionnye bazy dannyh: prakticheskie priemy optimal'nyh reshenij [Relational databases: practical techniques for optimal solutions]. Saint Petersburg: BHV, 2005. 392 p.
5. Struzhkin N., Godin V. Bazy dannyh. Proektirovanie: uchebnik [Databases. Design]. Mjscow: Jurajt, 2017. 478 p.
6. Codd, E.F. A Relational Model of Data for Large Shared Data Banks. *IBM Research Laboratory*. San Jose, California: Febrary Communications of the ACM (CACM), 1970.
7. Codd E.F. Relational. Completeness of Data Base Sublanguages. *Data Base Systems, Courant Computer Science Symposia Series 6*. Englewood Cliffs, N.S.: Prentice-Hall, 1972.
8. Codd E.F. The relation model for Database Model. Addison-wesley Pub-lishing Company, 1990. 540 p.

СВЕДЕНИЯ ОБ АВТОРЕ:

В.Ю. Пирогов, кандидат физико-математических наук, профессор кафедры программирования и автоматизации бизнес-процессов, ФГБОУ ВО «Шадринский государственный педагогический университет», г. Шадринск, Россия, e-mail: pirogov@shadrinsk.net.

INFORMATION ABOUT THE AUTHOR:

V.Y. Pirogov, Ph. D. in Physical and Mathematical Sciences, Professor, Department of Programming and Automation of Business Processes, Shadrinsk State Pedagogical University, Shadrinsk, Russia, e-mail: pirogov@shadrinsk.net.