

Владислав Юрьевич Пирогов
г. Шадринск

О некоторых аспектах преподавания основ реляционных баз данных в высших и средних учебных заведениях

В статье обсуждаются некоторые проблемы обучения проектированию реляционных баз данных студентов высших и средних учебных заведений. Анализируется вопрос описания предметной области на основе некоторых структур данных, в частности на основе совокупности заполненных электронных таблиц. Автор рассматривает случаи представления данных электронными таблицами, с точки зрения типов связи между сущностями, а также выполнения требований нормальных форм. Во второй части статьи автор описывает разработанную им программу, позволяющую ускорить разработку баз данных на основе электронных таблиц, которая может быть использована на занятиях по проектированию баз данных для более глубокого понимания проблем, возникающих на уровне анализа предметной области. Программа позволяет учитывать связи между таблицами, а также проводить декомпозицию таблиц для выполнения требования нормальных форм.

Ключевые слова: реляционные базы данных, СУБД, нормальные формы, проектирование баз данных, преподавание баз данных, декомпозиция.

Vladislav Yurievich Pirogov
Shadrinsk

Aspects of teaching the basics of relational databases in higher and secondary educational institutions

The article discusses some of the problems of teaching relational database design to students of higher and secondary educational institutions. The issue of describing the subject area is analyzed on the basis of some data structures, in particular, on the basis of a set of completed spreadsheets. The author examines the cases of data representation by spreadsheets, in terms of the types of relationships between entities, as well as meeting the requirements of normal forms. In the second part of the article, the author describes a program he has developed to speed up the development of spreadsheet-based databases, which can be used in database design classes to gain a deeper understanding of the problems that arise at the level of domain analysis. The program allows to take into account the relationships between tables as well as decompose tables to meet the requirements of normal forms.

Keywords: relational databases, DBMS, normal forms, database design, database teaching, decomposition.

Введение. Современные проблемы разработки информационных систем, в том числе и систем, в которых используется искусственный интеллект, тесно связаны с хранением данных, а, следовательно, требуют знаний в области структур данных, которые формируются на базе той или иной модели. До сих пор при разработке систем автоматизации той или иной области, превалирует использование реляционной модели данных [1, 2, 3, 6, 7, 8]. Следовательно, важным требованием к современному разработчику является наличие компетенции описания предметной области с помощью объектов реляционной модели. Такое описание предполагает глубокое понимание предметной области и умения отображать её в область реляционных баз данных [3, 4, 5]. Традиционно курсы по информационным системам, базам данных строятся на описании основных понятий реляционных баз данных (далее РБД), изучении языка SQL и практического его использования. Изучаются также другие объекты РБД уже относительно конкретных систем управления базами данных (далее СУБД). Также рассматриваются вопросы, связанные с таким разделом реляционной теории, как нормальные формы [3, 4]. Одной из задач, которая решается в подобных курсах, является задача проектирования

базы данных на основе описания предметной области. Как правило — это словесное описание, в котором выделяются основные сущности и отношения между ними. Это важная компетенция для будущих разработчиков информационных систем. Однако часто упускается задача, которую приходится решать специалистам в области разработки баз данных. А именно, частью описания предметной области могут являться те или иные структуры данных, которые были использованы ранее. Это могут быть отдельные таблицы в той или иной СУБД, данные в виде текстовых документов, в виде файлов csv, а также данные в виде электронных таблиц. Последнее встречается довольно часто. В данной статье мы рассмотрим вопрос об использовании электронных таблиц в качестве части описания предметной области как задачу, которая может быть применена в учебном процессе в качестве средства обучения студентов по специальностям, связанным с подготовкой специалистов в области разработки программного обеспечения. Мы также предлагаем разработанное нами приложение, позволяющее ускорить разработку и одновременно акцентировать внимание обучаемых на отдельных вопросах, связанных с проектированием.

Постановка задача. Стандартной задачей, которая ставится перед студентами, изучающими дисциплины, связанные с реляционными базами данных (Базы данных, Информационные системы, Разработка информационных систем и др.) является задача проектирования реляционной базы данных на основе словесного описания предметной области. Приведём пример такой задачи.

Спроектировать реляционную базу данных на основе предметной области «Единый классный журнал.» Учесть: классы, ученики, предметы, оценки, учителя.

Это краткое описание предметной области, дающее возможность студенту самому продумывать глубину, на которую он будет проектировать свою базу данных, а также продемонстрировать своё понимание того, как может функционировать информационная система на основе его базы данных. По сути, в самом задании уже перечислены все сущности, которые студент должен учесть при его выполнении. В такой постановке задачи нет проблем, связанных с неточностью в определении сущности, когда частью её может оказаться другая сущность и тем может быть нарушено требование одной из нормальных форм [1, 3]. Основная проблема здесь в правильном определении связей между таблицами (сущностями) и в определении в каждой паре связи главной и подчинённой таблицы.

Но предметная область может быть задана и другим способом, например, уже готовыми данными. Перед студентом может быть поставлена, например, такая проблема. Необходимо спроектировать базу данных на основе набора некоторого набора электронных таблиц. Фабула задачи может быть такой:

Работники некоторого отдела долгое время использовали электронные таблицы для хранения данных и автоматизации некоторых видов работ. Но пришло время автоматизировать работу с помощью некоторой информационной системы. Для этого следует проанализировать предметную область на основе набора электронных таблиц и спроектировать реляционную базу данных.

Конечно, в реальных условиях, разработчик имеет возможность исследовать не только имеющийся набор электронных таблиц, но и регламентирующие документы, а также анализировать работу отдела, что называется «вжувую»: беседовать с работниками, наблюдать за их работой, проводить опросы, анкетирование и т. д. Но анализ электронных таблиц и других форм хранения данных, используемых работниками отдела, наиболее интересен, с точки зрения обучения будущих специалистов.

Рассмотрим пример задания для студентов.

Пусть имеется всего одна электронная таблица, которая использовалась учителем математики для выставления оценок. В таблице имеются следующие колонки: ФИО ученика, Дата, Класс, Оценка. Учитель выставлял оценки для контроля, но со временем таблица стала большой и работать с ней стало трудно. Необходимо на основе этой таблицы спроектировать реляционную базу данных, выделив нужные сущности.

Замечательно то, что в этой задаче можно продемонстрировать сразу два подхода в получении корректной реляционной базы данных: использовать довольно формальные требования нормальных форм и подход, основанный на выделении сущностей. Оба подхода приводят к нужному результату. А именно, мы получаем три сущности или три таблицы, связанные между собой: Ученик, Класс, Оценки. На следующем этапе мы определяемся с тем, какие таблицы связаны между собой и с типом связи. Если тип связи «один ко многим», то здесь следует определиться, какая из таблиц главная, а какая подчинённая. При этом тип связи «один к одному» следует считать частным случаем связи «один ко многим». Очевидно, что в паре Ученик — Оценки, главной таблицей является таблица Ученик. Что касается пары Ученик и Класс, то здесь могут быть два возможных варианта. В первом случае, главной является таблица Класс, во второй — таблицы равноправны и связь здесь «многие ко многим», что приводит к необходимости использовать ещё одну таблицу, реализующую эту связь. Это несколько усложняет структуру базы данных, но позволяет, например, сохранить историю перехода ученика из класса в класс и оценок, которые он получал в каждом классе.

Представленный выше пример демонстрирует возможности проектирования баз данных, на основе уже существующих данных в виде электронных таблиц. Рассмотренную выше таблицу можно анализировать и с помощью технологии нормальных форм, и путём выделения в ней сущностей. Результат представлен на рисунке 1, где изображена схема базы данных, состоящей из трёх таблиц, в которой связь между ними представлена согласно положению реляционной теории: **первичный ключ-внешний ключ**. Также, на наш взгляд, важно требование перевода всех русских названий (столбцов, таблиц и базы данных) в транслит.

Вторая ситуация возникает, когда имеется несколько электронных таблиц, которые можно считать самостоятельными сущностями и которые, возможно, связаны друг с другом. Основная задача, которая ставится перед студентами, это определить эти связи (рис. 1).

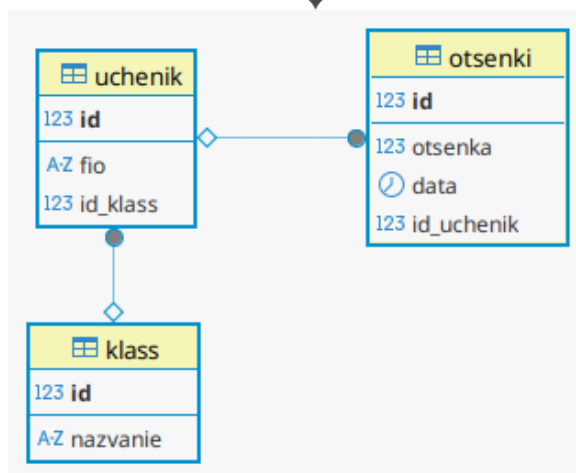


Рис.1. Схема данных после разделения сущностей

Рассмотрим следующий пример. Имеются две электронные таблицы «Работники» со столбцами ФИО, Должность, Адрес и таблица «Зарплата» со столбцами ФИО, К выдаче и Дата. Задание: на основе данных электронных таблиц разработать реляционную базу данных. Разумеется, сразу бросается в глаза столбцы ФИО в обеих таблицах. Опуская тот факт, что «фамилия, имя, отчество» далеко не всегда можно выбирать в качестве идентификатора (в курсе реляционных баз данных подробно разбирается вопрос, почему естественный ключ чаще всего не подходит в качестве первичного ключа [3]), можно поставить перед студентами задачу: определить точный критерий того, что две указанные таблицы реально связаны друг с другом по указанному столбцу. Если критерий выполняется (о критериях см. следующий раздел статьи), то далее всё сводится к переносу таблиц в СУБД и связывании их по технологии «первичный - внешний ключи». При корректном анализе и переносе получаем следующие две таблицы: `rabotniki<id, fio, dolznost, adres>` и `zarplata<id, zarplata, data, id_rabotniki>`. Связь между таблицами, таким образом, осуществляется через первичный ключ таблицы «rabotniki» `id` и внешний ключ `id_rabotniki` таблицы «zarplata». Следует лишний раз подчеркнуть, что: 1. Одинаковое название столбцов в двух таблицах не является признаком того, что именно по этому столбцу осуществляется связь. 2. Столбец ФИО удаляется из подчинённой таблицы и заменяется внешним ключом `id_rabotniki`.

Третий возможный вариант, когда база данных должна строиться на основе электронных таблиц, представляет собой смесь первой и второй ситуации, когда электронная таблица может выражать собой более одной сущности и, одновременно, возможны связи между такими сущностями (таблицами). Здесь следует определить последовательность действий: 1. Разбиваем электронные таблицы на несколько таблиц (осуществляем декомпозицию), если того требует выполнение условий нормальных форм. 2.

Проверяем возможные связи между таблицами и формируем базу данных.

Программное решение

Для обучения процессу анализа и проектирования баз данных на основе набора электронных таблиц, нами было разработано программное обеспечение (ПО). Данное ПО может быть использовано и как инструмент создания баз данных на основе набора электронных таблиц в полуавтоматическом режиме, и как инструмент обучения студентов. Данная программа пока реализована для СУБД PostgreSQL (см. [9]), в дальнейшем список используемых СУБД будет расширен. Остановимся на некоторых возможностях разработанного ПО и его использования в учебном процессе.

Основная функция программы заключается в запуске мастера, позволяющего в три шага сформировать базу данных на основе набора электронных таблиц (xlsx). При этом в сети или локально должен существовать сервер PostgreSQL, к которому программа при её запуске подсоединяется под ролью, у которой имеются права на создание баз данных и других реляционных объектов. Если у роли таких прав нет, то часть функциональности программа не сможет реализовать.

Работа мастера разбивается на три этапа: загрузка выбранных электронных таблиц, создание базы данных, редактирование связей и полный перенос таблиц со связями и содержимым в созданную базу данных. Следует отметить, что программа работает в полуавтоматическом режиме, но на каждом шаге пользователь может изменить элементы сценария: изменить структуру таблицы, разбить таблицу на несколько таблиц (осуществить декомпозицию), добавить или удалить связи. За невозможностью дать полное описание всего потенциала программы, мы остановимся на некоторых её возможностях. В окне каждого шага справа имеется секция, где подробно описывается каждый шаг (рис. 2).

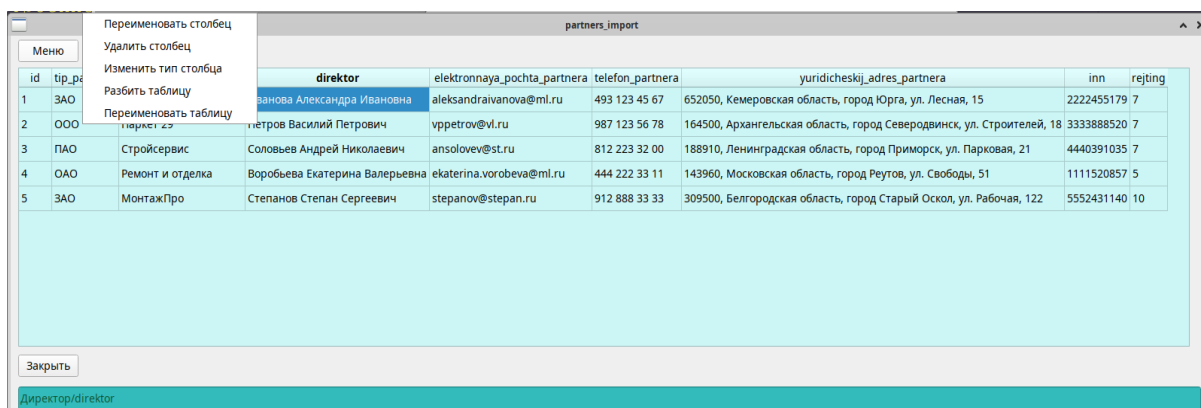


Рис. 2. Окно управления таблицей

При загрузке указанных таблиц в память, программа преобразует все имена с помощью прописных букв транслита, игнорируя различные не алфавитные знаки, удаляя лишние пробелы и заменяя оставшиеся знаками подчёркивания. При этом происходит анализ типов данных в столбцах и, при необходимости, осуществляется преобразование к единому типу. Эти операции производятся автоматически и важно указать студентам на необходимость такого начального преобразования и объяснить используемые алгоритмы. Важно также отметить, что программа автоматически добавляет в таблицу уникальный идентификатор (суррогатный первичный ключ), тем самым для таблицы автоматически будет выполняться требование второй нормальной формы [3].

В списке загруженных таблиц, можно выбрать любую из них, посмотреть её содержимое и выполнить с ней ряд операций. В меню (см. Рисунок 2) мы видим перечень этих операций. Обращаем, в частности, внимание на операцию Разбить таблицу. Она как раз и предназначена на случай необходимости приведения таблицы к

требование 3-5 нормальных форм, которые решаются разбиением (декомпозицией) таблицы, по некоторому столбцу. При этом возможно потребуется в одной из таблиц убрать дублирующие строки. Мы, таким образом, получим две связанные по столбцу таблицы, связь между которыми на последнем этапе обработки будет преобразована уже в классическую связь по линии первичный-внешний ключи. Важно отметить, что разбиение конкретной таблицы на две может и не закончить процесс декомпозиции. Тогда нужно повторить операцию декомпозиции над её частями. Заметим также, что при необходимости программа добавляет в таблицу уникальный идентификатор в качестве будущего первичного ключа.

При переходе к следующему этапу программа спрашивает студента имя базы данных, в которой будут созданы таблицы. Имя базы данных также преобразуется в транслит, кроме того, программа учитывает возможные совпадения имён баз данных. У студента есть возможность посмотреть имена баз данных, которые есть на данном сервере (рис. 3).

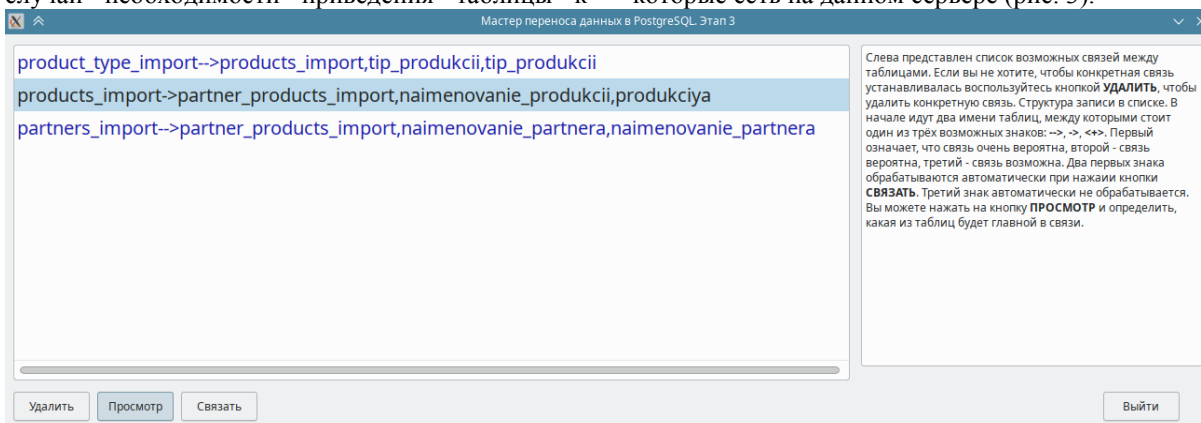


Рис. 3. Список с найденными связями между таблицами

Следующее окно (после создания базы данных) похоже на первое и у студента остаётся возможность проводить те же операции над таблицами, что и в предыдущем случае.

Окно третьего шага относится к связям (см. Рисунок 3). Программа анализирует все таблицы, сравнивая их друг с другом по каждому из столбцов, определяя вероятные связи. Одним из признаков, но не главным,

является одинаковые имена столбцов. Далее происходит сравнение содержимого столбцов. Здесь могут быть несколько вариантов. Вариант, позволяющий практически однозначно сказать, что таблицы связаны по указанным столбцам, заключается в следующем. В одной таблице, содержащее данного столбца не имеет дублей. Во второй таблице содержимое столбца состоит только из элементов столбца первой таблицы. При этом либо в столбце есть дубли, либо в нём присутствуют не все элементы из столбца первой таблицы. Описанный вариант не требует модификации со стороны обучаемого, но, если он не хочет, чтобы данная связь была реализована в базе данных, он может удалить её из списка (см. Рисунок 3). Следующий вариант, который рассматривает программа, когда столбца идентичны по содержимому и в них нет дублей. В этом случае программа не знает какая из таблиц будет главной в связи. Учащийся может открыть окно просмотра связи и в нём указать какая из таблиц главная. Ещё один случай отмечается программой, когда содержимое столбцов состоит из одинаковых элементов, но в обоих столбцах есть повторения. В этом случае связь между ними может быть осуществлена через третью таблицу. Это можно сделать после того, как база данных будет полностью сформирована на сервере с помощью того или иного инструмента. В программе будет добавлена такая функциональность в следующих версиях.

После того, как учащийся разобрался со связями, можно нажать на кнопку Связать (см. Рисунок 3), после этого содержимое таблиц, их структура и связи будут перенесены на сервер. При этом связи будут реализованы с помощью внешнего ключа, а столбец в подчинённой

таблице, указывающий на связь при анализе, будет удалён.

Как мы уже отмечали, программа таким образом, может быть использована как инструмент и как средство обучения. Такие темы как: первичный ключ, нормальные формы, связи между таблицами, анализ предметной области можно изучать с использованием данного программного продукта. Продукт прошёл экспериментальную апробацию на занятиях. Мы предполагаем дальнейшее развитие программы по нескольким направлениям: расширить возможности программы для работы с нормальными формами, добавить возможность программно создавать таблицу для связи между двумя другими таблицами, реализовать графическую визуальную схему связи между таблицами. Продукт разработан на языке программирования Python 3.9.

Заключение. В статье нами были рассмотрены вопросы преподавания основ реляционных баз данных. Был сделан акцент на проектирование баз данных на основе данных о предметной области. Нами было отмечено, что при обучении студентов проектированию, следует варьировать способы описания рассматриваемой предметной области. В частности, рассматривать вопрос, часто встречаемый на практике, когда основой описания предметной области является набор электронных таблиц, который использовался в работе тех или иных специалистов. Нами было разработано программное обеспечение, описанное в статье, которое позволяет ускорить процесс проектирования реляционной базы данных и одновременно обучить студентов анализу данных, представленных в электронных таблицах

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. – 8-е изд. – Москва : Вильямс, 2005. – 1328 с. – Текст : непосредственный.
2. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегт. – 3-е изд. – Москва : Вильямс, 2003. – 1436 с. – Текст : непосредственный.
3. Пирогов, В. Ю. Информационные системы и базы данных : учеб. пособие / В. Ю. Пирогов. – Санкт-Петербург : БХВ, 2009. – 528 с. – Текст : непосредственный.
4. Пирогов, В. Ю. Некоторые вопросы преподавания основ реляционных баз данных / В. Ю. Пирогов. – Текст : непосредственный // Вестник Шадринского государственного педагогического университета. – 2023. – № 4 (60). – С. 119–123.
5. Стружнин, Н. Базы данных. Проектирование : учебник / Н. Стружнин, В. Годин. – Москва : Юрайт, 2017. – 478 с. – Текст : непосредственный.
6. Codd, E. F. The relation model for Database Model = Реляционная модель для баз данных / E.F. Codd. – Boston : Addison-wesley Publishing Company, 1990. – 540 p. – Text : direct.
7. Codd, E. F. A Relational Model of Data for Large Shared Data Banks / E. F. Codd. – Text : direct // IBM Research Laboratory. San Jose, California: February 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 / E. F. Codd. – Addison-wesley Publishing Company, 1990. – 540 p. – Text : direct.

9. The PostgreSQL Global Development Group. PostgreSQL 17.4 Documentation / The PostgreSQL Global Development Group – Bellingham, Washington : 2025. – 3000 p. – Text : direct.

◆

REFERENCES

1. Date, K. J. (2005), *Introduction to database systems*, 8th ed., Moscow : Williams, 1328 p. (in Russian)
2. Connolly, T. and Begg, K. (2003), *Databases. Design, implementation and maintenance. Theory and practice*, 3rd ed., Moscow : Williams, 1436 p. (in Russian)
3. Pirogov, V. Y. (2009), *Information systems and databases: a textbook*, St. Petersburg: BHV, 528 p. (in Russian)
4. Pirogov, V. Y. (2023), 'Some issues of teaching the basics of relational databases', *Bulletin of Shadrinsk State Pedagogical University*, no. 4 (60), pp. 119–123. (in Russian)
5. Struzhkin, N. (2017), *Databases. Designing* : textbook, Moscow: Yurayt, 478 p. (in Russian)
6. Codd, E.F. (1990), *The relation model for Database Model*, Boston: Addison-wesley Publishing Company, 540 p. (in Russian)
7. Codd, E. F. (1972), *A Relational Model of Data for Large Shared Data Banks*, *IBM Research Laboratory*. San Jose, California: February 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. (1990), *The relation model for Database Model*, Addison-wesley Publishing Company, 540 p.
9. The PostgreSQL Global Development Group (2025), *PostgreSQL 17.4 Documentation*, The PostgreSQL Global Development Group, Bellingham, Washington, 3000 p.

СВЕДЕНИЯ ОБ АВТОРАХ:

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: vladislav-133@yandex.ru, ORCID:0009-0001-0874-3162.

INFORMATION ABOUT THE AUTHOR:

V.Y. Pirogov, Ph.D. in Physico-Mathematical Sciences, Professor, Department of Programming and Automation of Business Processes, Shadrinsk State Pedagogical University, Shadrinsk, Russia, e-mail: vladislav-133@yandex.ru, ORCID:0009-0001-0874-3162.