

Г. А. Онопенко, Н. А. Вихорь

# БАЗЫ ДАННЫХ

УЧЕБНОЕ ПОСОБИЕ

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Томский государственный архитектурно-строительный университет»

**Г.А. Онопенко, Н.А. Вихорь**

# **БАЗЫ ДАННЫХ**

Учебное пособие

Томск  
Издательство ТГАСУ  
2019

УДК 004.65(075.8)

ББК 32.97я73

**Онопенко, Г.А.** Базы данных [Текст] : учебное пособие /  
059 Г.А. Онопенко, Н.А. Вихорь. – Томск : Изд-во Том. гос. архит.-  
строит. ун-та, 2019. – 104 с.

ISBN 978-5-93057-908-6

В учебном пособии рассматриваются основные понятия баз данных. Приведены примеры выполнения практических заданий. Для закрепления полученных навыков в пособии приведены контрольные вопросы и индивидуальное задание.

Пособие предназначено для студентов, обучающихся по направлению 09.03.03. «Прикладная информатика» и изучающих дисциплины «Информационные системы и технологии», «Базы данных», а также для студентов направления 21.03.02 «Землеустройство и кадастры», изучающих дисциплину «Основы проектирования и создания баз данных».

**УДК 004.65(075.8)**

**ББК 32.97я73**

### **Рецензенты:**

**Н.Д. Сосновский**, канд. физ.-мат. наук, доцент кафедры прикладной математики ТГАСУ;

**А.Е. Петелин**, канд. физ.-мат. наук, доцент кафедры информационного обеспечения инновационной деятельности НИ ТГУ.

Редактор Т.С. Володина

Оригинал-макет подготовлен Н.А. Вихорь

Подписано в печать 23.12.2018.

Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.

Уч.-изд. л. 5,47. Усл. печ. л. 6,05. Тираж 15 экз. (1-й з-д). Зак. № 235.

Изд-во ТГАСУ, 634003, г. Томск, пл. Соляная, 2.

Отпечатано с оригинал-макета в ООП ТГАСУ.

634003, г. Томск, ул. Партизанская, 15.

ISBN 978-5-93057-908-6

© Томский государственный  
архитектурно-строительный  
университет, 2019

© Онопенко Г.А.,  
Вихорь Н.А., 2019

## ОГЛАВЛЕНИЕ

<b>Введение</b> .....	4
<b>1. Основы реляционных баз данных</b> .....	5
1.1. Терминология .....	5
1.2. Понятие реляционного отношения .....	9
1.3. Определение связей между таблицами .....	13
1.4. Классификация отношений.....	14
1.5. Основные функции реляционной СУБД .....	16
<b>2. Работа в LibreOffice Base</b> .....	21
2.1. Создание базы данных, состоящей из двух таблиц .....	22
2.2. Модификация базы данных .....	38
2.3. Создание базы данных, состоящей из трех таблиц .....	47
2.4. Создание и использование простых запросов.....	56
2.5. Использование других источников данных .....	67
2.6. Создание SQL-запросов .....	72
2.7. Создание отчетов .....	81
2.8. Создание итогового отчета .....	89
2.9. Кнопочная форма .....	93
<b>Контрольные вопросы</b> .....	102
<b>Индивидуальное задание</b> .....	103
<b>Список рекомендуемой литературы</b> .....	104

## ВВЕДЕНИЕ

XXI век – век информации, которая окружает нас и является неотъемлемой частью нашей повседневной жизни. Потоки и количество информации возросли настолько, что современная жизнь немыслима без эффективного управления информацией. Для этого используются системы обработки информации, в том числе основанные на базах данных (БД).

*База данных* – это организованная определенным образом, структурированная информация, для управления которой используются специальные программные средства – СУБД.

СУБД – система управления базами данных – это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования базы данных многими пользователями.

Данное учебное пособие позволяет студенту освоить одну из популярных открытых СУБД LibreOffice Base, которая поддерживает реляционную модель данных. В учебном пособии даются краткие теоретические сведения о реляционной модели данных и на практических примерах реальной базы данных показано поэтапно её создание, что позволяет изучить основные возможности работы с программой Base.

# 1. ОСНОВЫ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Подавляющее большинство БД поддерживают реляционную модель данных. Это связано со многими обстоятельствами. С точки зрения пользователя, реляционным базам данных свойственна наглядность и привычность представления данных: организация информации об объекте исследования в табличном виде использовалась задолго до включения ЭВМ в процесс ее обработки. Следовательно, у пользователей наработан опыт реляционного структурирования данных.

Однако самый важный аргумент в пользу реляционных баз данных состоит в том, что к началу 70-х гг. XX в. была подготовлена математическая основа манипуляции такими данными. Речь идёт о совокупности правил, согласно которым над данными с реляционной моделью можно выполнять вполне определенные операции и в результате этих операций извлекать новые знания об объекте исследования. Фактическая сторона вопроса манипуляции данными очень непростая, и пока она не была подготовлена математиками П. Коном, Дж. Фон Нейманом, М.Э. Мароном, Э.Ф. Коддом и некоторыми другими, автоматизация организационно-распорядительного управления (на западе – автоматизация деловых контор) была скорей мечтой, чем реальностью. Именно благодаря Э.Ф. Кодду, сотруднику ИВМ, который опубликовал в 1970 г. исследования в области алгебры отношений (отношение – relation – реляция), реляционная модель получила компьютерную жизнь. Кодд показал, что набор отношений (таблиц) может быть использован для хранения данных об объектах реального мира и моделирования связей между ними.

## 1.1. Терминология

Реляционная база данных – это набор взаимосвязанных *отношений*. Например, база данных высшего учебного заведения «ВУЗ» может включать следующие отношения: «Факультеты», «Кафедры», «Преподаватели», «Студенты» и т. д. Каждое из этих

отношений описывает отдельный логически завершённый фрагмент реальности (или, по-другому, предметная область), а их совокупность с учётом взаимосвязей представляет единый объект моделирования – ВУЗ.

Поскольку БД содержит множество отношений, каждое из них должно иметь собственное имя, чтобы обеспечить доступ к конкретному элементу множества. Имя отношения обязательно отражает смысл собранных в нём данных. По форме отношение – это таблица (при соблюдении определенных ограничений), а по содержанию – набор однотипных записей, соответствующих имени отношения.

При распределении информации об объекте моделирования по отдельным таблицам следует придерживаться ряда простых правил, которые наработаны практикой создания реальных баз данных.

Каждое отношение (таблица) должно содержать информацию только на одну тему. Сведения по разным темам обрабатываются намного легче, если они содержатся в независимых друг от друга таблицах. Например, данные о преподавателях и читаемых дисциплинах следует хранить в разных таблицах, с тем чтобы при удалении, например, преподавателя информация о дисциплине осталась в БД.

Сведения в таблицах не должны дублироваться: когда определенная информация хранится только в одной таблице, то и изменять ее придется только в одном месте. Это делает работу более эффективной и исключает возможность несоответствия информации в разных таблицах. Очевидно, адреса и телефоны преподавателей следует хранить в одной таблице.

Рассмотрим отношение СТУДЕНТ (табл. 1) и введем на этом примере основные понятия. В силу исторических причин почти каждый термин имеет несколько синонимов, которые можно встретить в разных источниках.

*Отношение – таблица – сущность*: таблично организованная информация, которую необходимо хранить в базе дан-

ных. В нашем примере это СТУДЕНТ.

*Запись – строка – кортеж – экземпляр сущности:* каждая строка таблицы, например,

21208 Волков А.А. 11.03.95 2 физика

*Таблица 1*

## СТУДЕНТ

№ Зачетной книжки	ФИО	Дата рождения	Курс	Специальность
21316	Иванов А.В.	23.11.95	2	биология
22121	Петров Г.С.	17.02.97	1	история
20118	Сидоров К.Т.	03.10.94	3	биология
21208	Волков А.А.	11.03.95	2	физика
19950	Антонов С.А.	02.06.93	4	экономика

*Поле – столбец – атрибут – свойство сущности:* имя каждого из столбцов таблицы, например, «Специальность» – поле отношения СТУДЕНТ. Основными характеристиками поля являются его *имя, тип хранимых данных и длина.*

*Значение атрибута – значение поля:* для заданного экземпляра таблицы содержание конкретного столбца, например, для экземпляра «Сидоров К.Т.» значение атрибута «Специальность» составляет «биология».

*Кардинальное число – количество записей в таблице.*

*Степень отношения – количество его столбцов (полей).*

*Схемой отношения* называют список имен всех полей с указанием типа данных и размера. Например, схема отношения СТУДЕНТ (№ Зачетной книжки, ФИО, Дата Рождения, Курс, Специальность). Для каждого поля следует указать один из допустимых типов (текстовый, числовой, дата...). Говорят, что отношения имеют *совместимые схемы*, если они имеют одинаковую степень и одинаковые типы соответствующих полей.

*Домен – множество всех допустимых значений для заданного поля.* Например, домен для поля «Курс» – это {1, 2, 3, 4, 5},



хотя конкретная таблица (табл. 1) использует только часть этого домена.

*Ключ отношения – первичный атрибут – ключевое поле* – особое поле, значения в котором однозначно определяют каждую запись таблицы. Отсюда и происходит сам термин – ключ «открывает» всю информацию о соответствующем ему экземпляре таблицы. Из определения следует, что значения ключевого поля для всех экземпляров таблицы *различны* в данный момент рассмотрения. В отношении СТУДЕНТ на роль ключа могут быть выбраны «ФИО», «Дата Рождения», «№ Зачетной книжки». Очевидно, что наилучшим идентификатором для экземпляров отношения СТУДЕНТ (ключом отношения) будет «№ Зачетной книжки», так как в общем случае фамилии у разных студентов могут совпадать, равно как и даты рождения. Для удобства ключ размещают в первом столбце таблицы.

Иногда ключ может состоять из нескольких атрибутов, тогда он называется *составным ключом*. Например, паспортные данные («Серия»+«Номер»). Каждый из атрибутов («Серия» или «Номер»), входящих в составной ключ, называется *частичным ключом*.

Дальнейшая классификация ключевых атрибутов связана с их использованием для организации межтабличных связей внутри одной и той же БД. Формальная связь одной таблицы (назовём её главной, родительской) с подчинённой ей (дочерней) организуется путём введения ключа дочерней таблицы в качестве дополнительного атрибута родительской таблицы. Тогда свой ключ естественно называть *первичным*, а сторонний, являющийся первичным в дочерней таблице, называют *вторичным (внешним) ключом* для родительской таблицы. Короче говоря, атрибуты, представляющие собой копии ключей других отношений, называются *внешними ключами*.

Например, в базе данных «ОПТОВАЯ БАЗА» есть таблица ТОВАРЫ (Код товара, Наименование, Поставщик, Цена, ...) и таблица ЗАКАЗЫ (Магазин, Код товара, Количество, ...). Здесь атрибут «Код товара» используется для объединения све-

дений о товарах и заказах из разных таблиц. В таблице ТОВАРЫ атрибут «Код товара» является первичным ключом, тогда как в таблице ЗАКАЗЫ этот же атрибут будет вторичным (внешним) ключом. Первичный ключ в таблице ЗАКАЗЫ – «Магазин». В схеме отношения первичный ключ обычно подчеркивают.

Таким образом, ключи бывают *простыми, составными, первичными и вторичными (внешними)*. В любом реляционном отношении *все неключевые атрибуты функционально зависят от первичного ключа*, а указание значения первичного ключа *однозначно определяет все свойства* соответствующего экземпляра таблицы.

## **1.2. Понятие реляционного отношения**

Использование формализма реляционной алгебры Кодда для вычисления новых отношений как результата выполнения операций над уже известными (старыми) отношениями требует соблюдения ряда ограничений при организации БД. Чем сложнее устройство БД и операции с её отдельными элементами, тем более жёсткие требования предъявляются к отношениям из БД.

Сформулировано шесть уровней строгости, или, как это называют в литературе, шесть нормальных форм для отношений: 1НФ (первая нормальная форма), 2НФ, 3НФ, нормальная форма Бойса – Кодда (НФБК), 4НФ, 5НФ. Каждая последующая все более требовательна к организации отношений, но вместе с тем все более эффективно использует аппарат алгебры отношений. Под эффективностью понимается минимизация дублирования данных и исключение функциональной зависимости между неключевыми атрибутами. На практике это значит, что обновление и обработка данных в одном месте БД не влечет коррекции данных в другом фрагменте БД. Метод достижения этой цели – декомпозиция (разложение) исходных отношений БД на другие, более мелкие и простые отношения. Понятно, что

при этом увеличивается число отношений в БД и тем самым увеличивается время ее обработки. С другой стороны, благодаря корректности и устранению дублирования данных ускоряется доступ к данным.

**Определение.** Отношение находится в первой нормальной форме, если все его атрибуты простые (атомарные).

Требование атомарности означает, что каждый атрибут отношения не разбивается на более мелкие элементы, т. е. не является ни списком, ни множеством значений.

Пример. Отношение СТУДЕНТ содержит сведения, представленные в табл. 2

Таблица 2

СТУДЕНТ

Фамилия	Курс	Специальность	Спорт	
			Вид	Разряд
Иванов	2	математика	плавание	м. с.
Петров	4	физика	футбол	к. м. с.
Сидоров	3	экономика	шахматы	1 разряд

Отношение СТУДЕНТ приводится к 1НФ путём декомпозиции сложного столбца «Спорт» на два атомарных: «Вид спорта» и «Спортивный разряд». Из примера заключаем, что *внешним признаком 1НФ отношения является простая шапка таблицы (её заголовок)*.

**Определение.** Отношение находится во 2НФ, если оно находится в 1НФ, и при этом все неключевые атрибуты зависят целиком от составного ключа, а не от какой-либо его части.

В качестве примера рассмотрим отношение ПРЕПОДАВАТЕЛЬ\_ПРЕДМЕТ (табл. 3) с составным ключом из двух атрибутов «Личный номер» и «Название предмета». Неключевой атрибут «Кол-во часов» зависит только от части ключа («Название предмета»), это и есть *частичная зависимость от ключа*.

Для того чтобы преодолеть неполную, частичную зависи-