

Базы данных

Оглавление

Краткие теоретические сведения	2
Реляционная база данных	4
Основные понятия	4
Ключи в реляционных базах данных	6
Многотабличные реляционные базы данных	6
Связи между таблицами. Понятие внешнего ключа	7
Системы управления базами данных	9
О терминологии	10
Проектирование баз данных и работа с ними	10
Проектирование баз данных	11
Основные операции с базами данных	17
Примеры решения заданий	21
Пример 1 задания с выбором правильного ответа	21
Пример 2 задания с одним верным ответом	22
Решения заданий демоварианта 2012	23
Задание А6	23
Характеристики задания	23
Задание	23
Решение	24
Задания для самостоятельного решения	25
Задание 1 с одним верным ответом	25
Задание 2 с одним верным ответом	25
Задание 3 с одним верным ответом	25
Задание 4 с одним верным ответом	26
Задание 5 с кратким ответом	26
Задание 6 с кратким ответом	26

Краткие теоретические сведения

Первые компьютеры создавались с целью автоматизации вычислений, решения сложных задач – расчета траектории полета ракеты, проектирования нового оборудования и т.п. Для решения каждой задачи разрабатывались алгоритмы и программы. Быстродействие первых компьютеров было невысоким, они были дорогими, их эксплуатация также требовала больших затрат.

С изменением элементной базы, появлением интегральных схем, стоимость компьютеров и их эксплуатации постепенно снижались, компьютеры стали доступны более широкому кругу пользователей. Появились средства долговременного хранения данных, стало развиваться второе основное направление использования компьютеров – хранение и обработка больших объемов информации. Проводились исследования по организации хранения данных, которая бы обеспечивала достаточную скорость доступа к данным, поиска необходимой информации, возможность пополнения данных и их изменения и т.д.

В 60-х годах XX века начались работы по созданию и развитию специализированных систем хранения и обработки данных, которые впоследствии стали называть системами управления базами данных (СУБД), а сами хранилища информации получили название «базы данных».

С базами данных в разных видах – книгах, картотеках – вы сталкивались уже не раз. Это телефонные справочники, словари, каталоги книг в библиотеках и т.д. Информация в них структурирована и упорядочена, то есть состоит из определенных данных и представлена в упорядоченном по какому-либо признаку виде.

Например, в телефонном справочнике данные упорядочены по фамилиям владельцев телефонов. Информация о книгах в библиотеках хранится на карточках, в которых записаны авторы, наименование книги, ее выходные данные, библиотечный шифр. Такие карточки объединяют в библиотечные каталоги, которые существуют в двух вариантах – упорядоченные по фамилиям авторов (алфавитный каталог) и по темам (тематический каталог).

Если же потребуется по телефонному справочнику в виде книги найти владельца телефона по номеру, придется просмотреть очень много страниц справочника.

Использование компьютеров для хранения баз данных позволяет хранить большие объемы структурированной информации, легко и быстро сортировать данные в любом порядке, осуществлять быстрый поиск информации, предоставлять информацию по запросам пользователей и т.д.

База данных (БД) – это совокупность структурированных взаимосвязанных данных о конкретных объектах реального мира, относящихся к определенной предметной области и хранящихся во внешней памяти компьютера

В определении БД используется термин «данные». Данные – это информация, зафиксированная в определенной форме и пригодная для обработки, хранения и передачи. У данных есть носитель, средства и методы их размещения.

База данных является **информационной моделью** предметной области.

Предметная область – часть реального мира, сведения о которой необходимо хранить, обрабатывать и использовать для решения конкретных задач (управления, автоматизации деятельности и т.п.

Предметная область состоит из объектов и связей между ними.

Объекты предметной области – это предметы, понятия, явления и т.д. Например, объекты предметной области «Библиотека» – это книга, читатель, хранилище (залы, стеллажи, полки), сведения о движении книг и т.д. Каждый объект имеет множество **свойств**. Не все свойства объектов важны для данной предметной области, поэтому из совокупности свойств объектов необходимо выделить наиболее существенные. Например, свойства объекта «книга» – автор, наименование, шифр, год издания важны для библиотеки, а сорт использованных при ее печати бумаги или типографской краски не важны. Существенные свойства объекта «читатель» – фамилия, имя, отчество, адрес, номер читательского билета, но не цвет глаз и не размер обуви.

Каждый объект предметной области состоит из экземпляров. **Экземпляры объектов** «книга» и «читатель» – это конкретная книга (например, «Информатика. 11 класс») и конкретный читатель (например, Колосов Сергей Петрович).

При разработке информационной модели следует не только выделить объекты предметной области, выбрать наиболее существенные свойства объектов, определить их взаимосвязи, но и выбрать модель данных.

Модель данных – это описание структур данных, хранимых в базе данных, и связей между ними¹

В настоящее время промышленным стандартом является табличная (реляционная) модель данных.

¹ Более полное определение модели данных включает операции обработки данных, но здесь они не рассматриваются

Реляционная база данных

Реляционная база данных основана на реляционной модели данных (англ. relation – отношение), которая была предложена Е. Коддом в 1970 году. Реляционная модель данных опирается на теорию множеств и математическую логику, что обеспечивает ее математическую строгость и обуславливает широкое распространение этой модели на практике.

Основные понятия

В реляционной БД данные хранятся в виде совокупности взаимосвязанных двумерных таблиц.

Каждая таблица реляционной БД описывает объект предметной области, имеет имя и состоит из строк и столбцов.

Столбцы таблицы соответствуют свойствам объекта и называются **полями**. Каждое поле имеет имя – заголовок столбца таблицы. В таблице не должно быть двух одинаковых имен полей. Каждое поле имеет определенный тип данных, т.е. все ячейки одного столбца таблицы должны содержать данные одного типа.

Напомним, что тип данных определяет

- 1) представление данных в памяти (в том числе сколько байт памяти данные будут занимать);
- 2) возможные значения данных;
- 3) допустимые действия над данными указанного типа.

В базах данных используются следующие типы данных: числовой (целое, действительное и т.д.), дата, текстовый, логический и т.д.²

Имена полей образуют **заголовок таблицы**. Он определяет структуру таблицы и не меняется во время работы с базой данных. Если в таблице изменить, добавить или удалить поля, то получим уже другую таблицу (пусть даже с прежним именем).

Строки таблицы называют записями. Каждая строка таблицы описывает один экземпляр объекта. В таблице не должно быть двух одинаковых записей.

Запись – это совокупность данных разного типа, относящихся к одному экземпляру объекта.

Тело таблицы – набор записей об экземплярах объекта. Тело таблицы изменяется во время работы с базой данных – записи об экземплярах объекта могут изменяться, добавляться и удаляться.

² В различных СУБД наименования типов могут различаться.

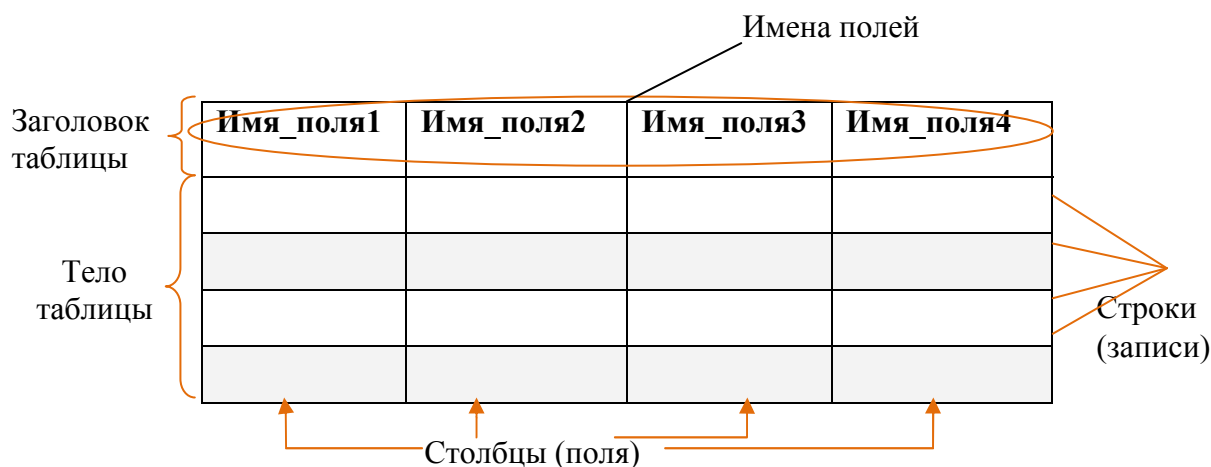


Рис. 9.1. Таблица реляционной модели данных

Приведем пример базы данных «Телефонный справочник». В справочнике должно храниться: номер телефона, фамилия, имя, отчество владельца телефона, адрес, состоящий из наименования улицы, номера дома (будем считать, что номер дома обозначается одним числом) и номера квартиры. Номер телефона должен быть закреплен строго за одним владельцем.

Если все эти данные поместить в одну таблицу, можно получить примерно следующий список полей:

Структура базы данных «Телефонный справочник»

Имя поля	Тип данных
Телефон	текстовый
Фамилия	текстовый
Имя	текстовый
Отчество	текстовый
Улица	текстовый
Дом	числовой целый
Кв	числовой целый

Заголовок таблицы имеет вид:

Телефонный справочник (Телефон, Фамилия, Имя, Отчество, Улица, Дом, Кв)

Заполненная таблица может иметь вид

Телефон	Фамилия	Имя	Отчество	Улица	дом	кв
225-64-90	Анастази	Ирина	Михайловна	1-я Первомайская	6	8
159-58-36	Байбурин	Александр	Борисович	Космонавтов	15	145
566-89-74	Власов	Олег	Петрович	Серебристый	130	75
198-34-54	Воеводин	Петр	Вадимович	1-ая Первомайская	10	28
328-35-75	Петров	Сергей	Сергеевич	Серебристый бульв.	1	12
225-64-91	Мишин	Михаил	Иванович	Первая Первомайская	10	81

Ключи в реляционных базах данных

Подобная база данных может иметь десятки тысяч записей (строк). Предположим, что оператор, заполняющий базу данных, по ошибке ввел дважды сведения об одном и том же номере телефона и его владельце. Через некоторое время владелец телефонного номера переехал в другой город, а телефонный номер перерегистрировали на имя другого владельца (изменили поля Фамилия, Имя, Отчество в соответствующей записи). Тогда может сложиться такая ситуация, что в одной записи базы данных оператор изменит информацию о владельце, а в другой – нет, и один телефонный номер будет зарегистрирован дважды – на фамилию прежнего и нового владельца. В базе появятся противоречивые данные. Для обеспечения непротиворечивости данных необходимо, чтобы номера телефонов в таблице БД были уникальными, то есть не повторялись. Значения других полей могут повторяться, например, названия улиц или фамилии.

Если будет обеспечена уникальность значений номеров телефонов, то появление в таблице второй строки с тем же номером телефона станет невозможно.

Совокупность полей, однозначно определяющих запись в пределах таблицы, называется **потенциальным ключом**. В таблице может быть несколько потенциальных ключей. Потенциальный ключ, состоящий из одного поля, называется **простым**. Потенциальный ключ, состоящий из нескольких полей, называется **составным**. Потенциальный ключ, выбранный основным, называется **первичным ключом** или просто ключом. Остальные потенциальные ключи называют альтернативными. В базе данных «Телефонный справочник», например, ключом может быть номер телефона. Альтернативным ключом может быть совокупность полей Фамилия, Имя, Отчество, Улица, Дом, Квартира. Как правило, в программах проектирования и работы с БД первичный ключ визуально выделяется в описании структуры таблицы, например, шрифтом или специальным значком.

Таблицы реляционных баз данных должны иметь первичный ключ

Многотабличные реляционные базы данных

Предположим, принято решение о переименовании улицы 1-я Первомайская. На улице могут находиться десятки домов, в каждом из которых – десятки квартир, в БД могут находиться сотни записей. Оператор должен внести изменения в поля всех записей, соответствующие 1-й Первомайской улице, иначе данные станут противоречивыми. При этом придется учитывать, что наименование улицы может быть внесено в БД в разных видах, и появление ошибок неизбежно.

Подобные проблемы возникли, т.к. мы смешали в одной таблице сведения о разных объектах предметной области: данные о номерах телефонов и наименования улиц. Одним из признаков того, что в таблицу включены данные о разных объектах, является повторение одних и тех же значений некоторых полей во многих строках таблицы. Это называют избыточностью данных. Выходом из такой ситуации является разделение таблицы базы данных на несколько таблиц, каждая из которых описывает объект предметной области.

Разделим таблицу БД «Телефонный справочник» на две связанные между собой таблицы – «Номера телефонов» и «Улицы». Для связи двух таблиц используются некоторые поля одного и того же типа в обеих таблицах. Это может быть целочисленный тип, строковый тип и т.п.

Связи между таблицами. Понятие внешнего ключа

В таблице «Улицы» не должно быть двух одинаковых строк, т.е. в этой таблице название одной и той же улицы не должно встречаться два и более раз. В таблице «Номера телефонов», напротив, названия одной и той же улицы могут встречаться многократно. В современных базах данных используют в основном связи типа «один ко многим». В нашем примере это означает, что одна улица из таблицы «Улицы» может много раз встречаться в таблице «Номера телефонов». Таблица «Улицы» называется родительской, таблица «Номера телефонов» - дочерней. В родительской таблице поле, по которому происходит связывание, должно быть ключевым. Это обеспечивает однозначность связи между таблицами.

Введем числовое целое поле «номер улицы» в таблицу «Улицы». Как правило, в качестве номера используется целочисленный тип данных «счетчик», имеющийся практически во всех СУБД. Отличие этого типа от обычного целочисленного заключается в том, что значения этого поля автоматически увеличивается на единицу всякий раз при добавлении новой записи в таблицу. После удаления записи из таблицы значение счетчика повторно не используется. Кроме того, пользователь не может изменить значение счетчика. Таким образом, значения поля типа «счетчик» не повторяются в таблице, и такое поле принято использовать в качестве первичного ключа.

Для обеспечения связи таблиц поле целочисленного типа, в котором хранятся значения «номер улицы», должно быть и в таблице «Номера телефонов», но здесь оно уже не будет иметь тип «счетчик», т.к. не является первичным ключом, значения номеров улиц могут повторяться.

Заголовки таблиц могут иметь вид:

Номера телефонов(Телефон, Фамилия, Имя, Отчество, номер_улицы, Дом, Кв);

Улицы(номер_улицы, Улица).

Заполненные таблицы могут иметь вид

Номера телефонов

Телефон	Фамилия	Имя	Отчество	номер_улицы	дом	кв
159-58-36	Байбурин	Александр	Борисович	2	15	145
198-34-54	Воеводин	Петр	Вадимович	1	10	28
225-64-90	Анастаси	Ирина	Михайловна	1	6	8
225-64-91	Мишин	Михаил	Иванович	1	10	81
328-35-75	Петров	Сергей	Сергеевич	3	1	12
566-89-74	Власов	Олег	Петрович	3	130	75

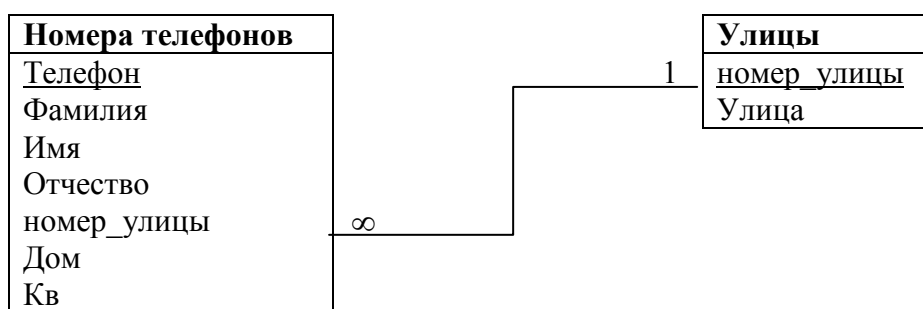
Улицы

номер_улицы	Улица
1	1-я Первомайская
2	Космонавтов
3	Серебристый бульвар

Обратите внимание на изменение порядка строк в таблице «Номера телефонов» - строки отсортированы по значению ключевого поля «Телефон».

Таблица «Улицы» содержит значительно меньше строк, т.к. количество улиц в разы меньше количества телефонных номеров. Теперь при переименовании улицы достаточно изменить ее название в таблице «Улицы».

Связь между таблицами показывается в **схеме базы данных**, состоящей из таблиц и связей между ними, которая может иметь вид:



Линия связи соединяет поля, по которым таблицы связаны. На концах линии могут находиться специальные значки. В этом примере показан тип связи «один ко многим». Это означает, что «на одной улице может быть зарегистрировано много номеров телефонов» и «один номер телефона может быть установлен только на одной улице». Напомним, что таблица, входящая в связь со стороны «один», называется родительской, со стороны «много» – дочерней.

Связь «один ко многим» реализуется добавлением в дочернюю таблицу поля (в этом примере «номер_улицы»), являющегося ссылкой на ключевое поле родительской

таблицы. Это поле называют мигрирующим из родительской таблицы, оно является **внешним ключом**. Значения внешнего ключа, в отличие от значений первичного ключа, могут повторяться. Имя внешнего ключа не обязательно совпадает с именем ключевого поля в родительской таблице, но тип внешнего ключа должен совпадать с типом первичного ключа в родительской таблице.

При работе с многотабличной БД может возникнуть ситуация, когда запись в родительской таблице удаляется по тем или иным причинам. Тогда может получиться, что некоторые записи дочерней таблицы ссылаются на несуществующее значение первичного ключа родительской таблицы. Возникает проблема обеспечения целостности по ссылкам. Как правило, в СУБД реализуются разные механизмы обеспечения целостности, например:

- 1) запрет удаления записи в родительской таблице, если на нее есть ссылка в дочерней;
- 2) при удалении записи в родительской таблице удаляются все записи в дочерних таблицах, ссылающихся на нее (каскадное удаление).

Системы управления базами данных

Для работы с базами данных: разработки таблиц и связей между ними, добавления, изменения, удаления, сортировки и поиска данных служат специальные программы – системы управления базами данных (СУБД). Существует множество разновидностей СУБД: от простых, с которыми может работать один или несколько пользователей, до очень сложных, на основе которых строятся системы автоматизации крупных организаций, когда сотни людей одновременно работают с огромными объемами самых разных данных.

СУБД – это специализированное программное обеспечение для создания и использования баз данных: редактирования, просмотра и поиска информации в них.

СУБД различаются по своим возможностям, но, как правило, обеспечивают выполнение следующих функций:

- создание новых БД (создание таблиц и связей между ними);
- заполнение БД;
- редактирование БД;
- сортировка данных;
- обработка запросов пользователей;
- поиск информации в БД;

- печать данных в форме отчетов и т.д.

Примеры реляционных СУБД: Microsoft Access, MS SQL Server, MySQL и другие. В школах изучается СУБД Microsoft Access.

О терминологии

В разных учебниках для 11 класса можно встретить разные термины, обозначающие казалось бы одни и те же понятия. Поясним эту ситуацию.

Взаимосвязь теории множеств и математической логики, реляционной модели данных и СУБД можно условно изобразить в виде следующей пирамиды:



Рис. 9.2

Каждый более высокий уровень основывается на понятиях, определенных на более низком уровне. На каждом из уровней используется своя терминология. Например, на уровне реляционной модели используются термины "отношение", "кортеж". На уровне СУБД используются термины "таблица", "строка таблицы".

Реляционная модель	СУБД
База данных	Набор таблиц
Схема базы данных	Набор заголовков таблиц с указанием связей между таблицами
Отношение	Таблица
Заголовок отношения	Заголовок таблицы
Тело отношения	Тело таблицы (набор строк)
Атрибут отношения	Наименование столбца таблицы или имя поля
Кортеж отношения	Строка таблицы или запись, состоит из значений атрибутов
Степень отношения	Количество столбцов (полей) таблицы
Мощность отношения	Количество строк (записей) таблицы

Проектирование баз данных и работа с ними

С базами данных работают три категории людей: проектировщики, администраторы и пользователи.

При проектировании БД необходимо определить, из каких таблиц будет состоять БД, какие поля и ключи будут в каждой таблице и как таблицы будут связаны между собой.

Администраторы баз данных поддерживают их в работоспособном состоянии.

Пользователи обращаются к БД в поисках нужной информации или с целью получения новой информации путем обработки данных, хранящихся в БД.

Проектирование баз данных

При проектировании баз данных выполняются следующие этапы:

- 1) анализ предметной области и выявление объектов, которые необходимо включить в базу данных. Каждый объект соответствует таблице БД;
- 2) определение свойств объектов и типов данных, описывающих эти свойства;
- 3) определение первичных ключей каждой таблицы;
- 4) определение связей между таблицами.

Пример 1. Проектирование базы данных «Детская поликлиника».

Ведется учет посещений пациентов с указанием ФИО пациента, пола, даты рождения, номера медицинской карточки, ФИО врача и его специализации, а также даты и времени посещения врача пациентом.

Разместим все данные в одной таблице:

Ном карт	ФИО пац	ДР пац	ФИО врач	Спец врач	Дата посещ	Время посещ
1105	Иванов И.П.	12.05.1998	Шокуров П.П.	Педиатр	08.08.2010	12:00
4021	Шаров Н.Х.	25.09.1998	Дарьина Е.П.	Кардиолог	08.08.2010	13:20
3251	Козлова Е.А.	12.09.1994	Шумов П.Д.	Педиатр	09.08.2010	09:30
1245	Гайдуков Б.А.	14.06.1991	Вострик П.П.	Стоматолог	08.08.2010	10:00
6874	Жуков А.Н.	24.05.1995	Гарибян Н.Р.	Педиатр	09.08.2010	10:00
1105	Иванов И.П.	12.05.1998	Дарьина Е.П.	Кардиолог	09.08.2010	10:00
4021	Шаров Н.Х.	25.09.1998	Шумов П.Д.	Педиатр	05.08.2010	15:20
3251	Козлова Е.А.	12.09.1994	Жульков В.В.	ЛОР	09.08.2010	11:30
1245	Гайдуков Б.А.	14.06.1991	Гильманов О.Р.	Стоматолог	08.08.2010	12:00
6874	Жуков А.Н.	24.05.1995	Гарибян Н.Р.	Педиатр	13.08.2010	11:00

При этом могут появиться проблемы противоречивости и избыточности данных, обеспечения целостности, описанные выше в примере «Телефонный справочник». Таблицу сложно обновлять и редактировать.

Кроме того, появляется еще одна проблема. Допустим, пациент Гайдуков переходит из детской поликлиники во взрослую. Если удалить все строки, содержащие

информацию о нем, то удалится информация и о врачах, которых он посещал. Таким образом, удаляя данные о пациентах, можно удалить и все данные о каком-либо враче. С другой стороны, если какой-то врач (например, главный) не ведет прием пациентов, то сведения о нем не попадут в базу данных.

Выделим объекты предметной области. Очевидно, сведения о пациентах и о врачах как о разных объектах должны храниться в разных таблицах. При этом в таблице пациенты ключом будет поле №_карт. В таблицу Врачи добавим поле-счетчик, которое будет ключом. Обычно имена таких специально введенных полей начинаются с символов ИД_ или ID_ (от слова «идентификатор»).



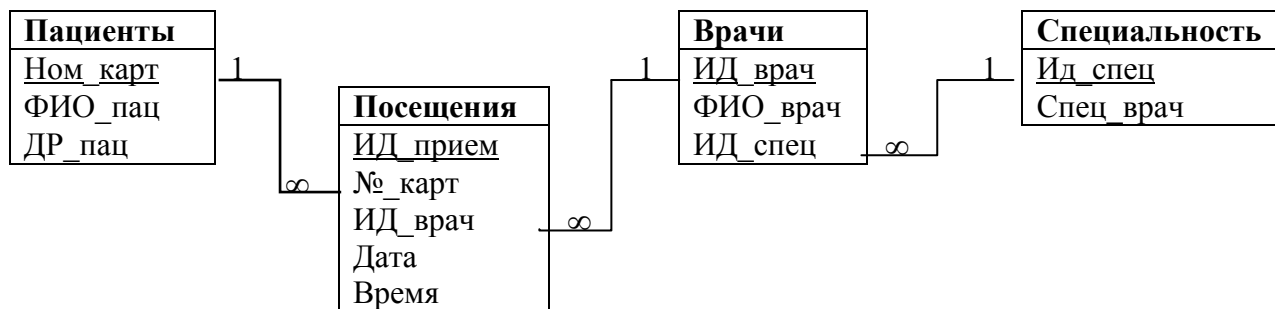
Пока непонятно, к какой таблице отнести дату и время приема. Попробуем установить вид связи между таблицами «Пациенты» и «Врачи». Очевидно, можно утверждать, что «многие пациенты посещают многих врачей». Такой вид связи называется «многие ко многим». Как правило, в СУБД она не реализуются. В базу данных вводится новая таблица, которая разбивает связь «многие ко многим» на две или более связей типа «один ко многим». При этом не следует забывать о включении в таблицу на конце связи «много» мигрирующего ключевого поля родительской таблицы.

Получим схему БД:



Напомним, что для каждого значения внешнего ключа должно существовать соответствующее значение первичного ключа в родительском отношении.

Заметим, что в таблице Врачи присутствует избыточность: повторяются значения поля Спец_врач. Очевидно, врачебная специальность – объект предметной области, и может быть вынесена в отдельную таблицу. Получим следующую схему БД:



Процесс разделения таблиц на несколько таблиц меньшего размера (декомпозиция) называется нормализацией отношений. Здесь не рассматриваются правила нормализации, дополнительные материалы по этой теме можно найти в учебниках.

Таблицы БД после декомпозиции имеют вид:

Пациенты

Ном_карт	ФИО_пац	ДР_пац
1105	Иванов И.П.	12.05.1998
1245	Гайдуков Б.А.	14.06.1991
3251	Козлова Е.А.	12.09.1994
4021	Шаров Н.Х.	25.09.1998
6874	Жуков А.Н.	24.05.1995

Врачи

ИД_врач	ФИО_врач	ИД_спец
1	Шокуров П.П.	1
2	Дарьина Е.П.	2
4	Шумов П.Д.	1
7	Вострик П.П.	3
12	Гарибян Н.Р.	1
15	Жульков В.В.	4
28	Гильманов О.Р.	3

Специальности

ИД_спец	Спец_врач
1	Педиатр
2	Кардиолог
3	Стоматолог
4	ЛОР

Таблица *Person*

<i>Наименование поля</i>	<i>Тип поля</i>
ID_Person	Счетчик
FIO	строка
Pol	строка

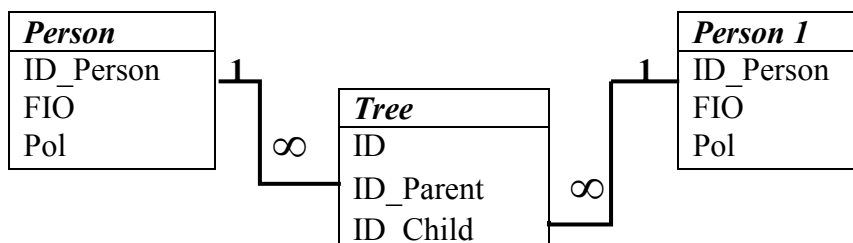
Кроме того, в отдельной таблице покажем родственные отношения, назовем ее *Tree*. В этой таблице каждая строка соответствует родственной связи двух людей – одного из родителей и одного из их детей.

Таблица *Tree*

<i>Наименование поля</i>	<i>Тип поля</i>
ID	Счетчик
ID_Parent	целое
ID_Child	целое

Очевидно, что данные в поля ID_Parent и ID_Child таблицы *Tree* должны подставляться из таблицы *Person* (поле ID_Person). Таким образом, должно быть две связи типа «один ко многим» между таблицами *Person* и *Tree*.

Обычно для установления двух связей между двумя таблицами в СУБД используют «копию» дочерней таблицы (на самом деле такая копия не создается, а используется второе имя дочерней таблицы, т.н. алиас). В СУБД схема данных для нашего примера может иметь вид:



Заполненные в соответствии с заданным фрагментом дерева таблицы могут иметь вид.

Person

<i>ID_Person</i>	<i>FIO</i>	<i>Pol</i>
12	Лысенко П.М.	М
25	Лысенко О.В.	Ж
41	Цветков М.В.	М
52	Петренко С.М.	Ж
64	Лысенко В.П.	М
101	Цветкова С.П.	Ж
121	Цветков Е.М.	М
155	Цветков Г.Е.	М
288	Дьяченко Г.Е.	Ж
304	Цветков П.Г.	М
325	Зац Л.А.	Ж
483	Федорова Т.А.	Ж

Tree

<i>ID</i>	<i>ID_Parent</i>	<i>ID_Child</i>
25	12	101
38	25	101
47	25	64
49	101	155
52	12	64
68	288	483
97	101	288
112	41	121
135	121	155
146	52	121
150	121	288
158	155	304
170	288	325

Допустим, необходимо определить по заданным фрагментам таблиц базы данных дедушку Федоровой Т.А.

1) Найдем в таблице **Person** строку с данными о Федоровой Т.А., определим ее **ID_Person** - 483

2) В таблице **Tree** найдем строки, в которых значение поля **ID_Child** равно 483, определим родителей Федоровой Т.А. – это человек с **ID_Parent** равным 288. Данных о другом родителе в таблице нет.

3) Теперь в таблице **Tree** необходимо найти родителей человека с номером 288. Эти люди имеют номера **ID_Parent**, равные 101 и 121.

4) По таблице **Person** определим, что под номером 101 записана женщина Цветкова С.П., т.е. бабушка Федоровой. Под номером 121 записан мужчина Цветков Е.М., дедушка Федоровой Т.А.

Таким образом, дедушкой Федоровой Т.А. является Цветков Е.М.

Основные операции с базами данных

Запросы на выборку данных

Обращение пользователя к БД с целью получения информации называется **запросом**. Для составления запросов используются специальные языки запросов. Чаще всего используются запросы на выборку данных, в качестве критерия отбора используются логические выражения. Логические выражения могут быть простыми и составными.

В простых логических выражениях накладываются требования на значение поля. Например, если в приведенной выше БД «Телефонный справочник» требуется найти владельцев телефонов с фамилией Волков, условие отбора будет иметь вид: «Фамилия = "Волков"».

Для построения запросов будем использовать базу данных «Туристические поездки». Структура таблицы базы данных:

Имя поля	Тип данных
Страна	Текстовый
Город	Текстовый
Продолжительность	Числовой целый
Стоимость	Числовой целый
Дата_заезда	Дата

Заполненная таблица:

Страна	Город	Продолжительность	Стоимость	Дата заезда
Россия	Москва	5	40 000	14.07.2010
Франция	Париж	7	65 000	14.07.2010
Россия	Сочи	18	70 000	12.06.2010
Великобритания	Лондон	8	70 000	15.07.2010
Италия	Турин	5	50 000	02.08.2010
Италия	Рим	7	65 000	02.09.2010

Для отбора туров продолжительностью от семи дней и больше условие отбора имеет вид: «Продолжительность ≥ 7 ».

Составные логические выражения конструируются из простых при помощи логических операций И, ИЛИ, НЕ. Например, в приведенной выше БД «Туристические поездки» необходимо найти туры стоимостью от 50 000 до 68 000. Условие отбора имеет вид: «Стоимость ≥ 50000 И Стоимость ≤ 68000 ». Этому условию отвечают три записи, в результате выполнения запроса получим таблицу:

Страна	Город	Продолжительность	Стоимость	Дата заезда
Франция	Париж	7	65 000	14.07.2010
Италия	Турин	5	50 000	02.08.2010
Италия	Рим	7	65 000	02.09.2010

Сортировка данных

Одной из основных операций при работе с базами данных является **сортировка**. При выполнении этой операции требуется указать поля, по которым проводится сортировка, и порядок сортировки (по возрастанию или по убыванию значений). Следует учитывать тип данных полей, по которым проводится сортировка.

Текстовые значения (строки) сортируются в лексикографическом порядке в соответствии с кодовой таблицей. Это значит, что две строки сравниваются посимвольно слева направо до первого несовпадающего символа.

Строка, в которой первый несовпадающий символ имеет больший номер в таблице кодировки символов, считается больше. Например, строка "Иванников" меньше строки "Иванов", т.к. первые несовпадающие символы – «н» и «о». Символ «н» расположен в кодовой таблице раньше символа «о», имеет меньший номер в кодовой таблице. Строка "СТУДЕНТ" меньше строки "студент", т.к. заглавные символы в кодовой таблице имеют меньшие номера, чем строчные.

Если строки имеют различную длину, но в общей части символы совпадают, более короткая строка меньше, чем более длинная, например, строка "Иванова" больше строки "Иванов".

Строки равны, если они полностью совпадают по длине и содержат одну и ту же последовательность символов.

Напомним порядок расположения символов в кодовой таблице ASCII:

- 1) Символы цифр;
- 2) Заглавные латинские буквы в алфавитном порядке,
- 3) Малые латинские буквы в алфавитном порядке,
- 4) Заглавные русские буквы в алфавитном порядке,
- 5) Малые русские буквы в алфавитном порядке.

Приведем пример сортировки строк. Заданы строки символов латинского алфавита

"ABCD", "Abcd", "abcd", "ABC", "ABc"

После сортировки по возрастанию они расположатся в следующем порядке:

"ABC", "ABCD", "ABc", "Abcd", "abcd"

После сортировки по убыванию:

"abcd", "Abcd", "ABc", "ABCD", "ABC".

Еще один важный пример иллюстрирует, что необходимо учитывать тип данных при выполнении сортировки. Допустим, в БД «Классные руководители» есть поле **Класс** и поле **Классрук**.

Класс	Классрук
2	Перова М.В.
8	Савельева Н.К.
11	Дедушкин П.В.
9	Егоров А.А.
1	Воинов А.Д.
10	Егорова К.Д.

Очевидно, поле **Классрук** имеет тип текстовый. А вот тип поля **Класс** может быть как числовым, так и текстовым. Рассмотрим сортировку БД в двух случаях:

- 1) поле **Класс** имеет тип числовое целое;
- 2) поле **Класс** имеет тип текстовый.

После проведения сортировки по возрастанию по полю **Класс** получим

тип поля Класс – числовое целое	
Класс	Классрук
1	Воинов А.Д.
2	Перова М.В.
8	Савельева Н.К.
9	Егоров А.А.
10	Егорова К.Д.
11	Дедушкин П.В.

тип поля Класс – текстовый	
Класс	Классрук
1	Воинов А.Д.
10	Егорова К.Д.
11	Дедушкин П.В.
2	Перова М.В.
8	Савельева Н.К.
9	Егоров А.А.

Если первая таблица не вызывает вопросов, то вторая требует пояснений. В кодовой таблице символы цифр расположены в следующем порядке '0', '1', '2', ..., '8', '9'. Для текстовых значений сравниваются не числа, а последовательности символов. Символ '0' расположен в кодовой таблице раньше символа '1', а символ '1' раньше символа '2', поэтому строка "10" меньше строки "11", а строка "11" меньше строки "2".

Записи (строки таблицы) сортируются целиком.

Отсортируем базу данных «Туристические поездки» по возрастанию стоимости.

Получим:

Страна	Город	Продолжительность	Стоимость	Дата заезда
Россия	Москва	5	40 000	14.07.2010
Италия	Турин	5	50 000	02.08.2010
Франция	Париж	7	65 000	14.07.2010
Италия	Рим	7	65 000	02.09.2010
Россия	Сочи	18	70 000	12.06.2010
Великобритания	Лондон	8	70 000	15.07.2010

Примеры решения заданий

Пример 1 задания с выбором правильного ответа

База данных о торговых операциях дистрибутора состоит из трех связанных таблиц. Ниже даны фрагменты этих таблиц:

Таблица зарегистрированных дилеров:

Наименование организации	ID дилера	Регион	Адрес
ООО «Луна»	101	Башкортостан	г. Уфа, проспект Октября, 15
АО «Венера»	102	Татарстан	г. Казань, ул. Булгар, 17
АОЗТ «Марс»	103	Марий Эл	г. Йошкар-Ола, просп. Мира, 8
ООО «Юпитер»	104	Чувашия	г. Чебоксары, ул. Волкова, 6
ИЧП Шакуров	105	Башкортостан	г. Нефтекамск, ул. Правды, 28
АО «Меркурий»	106	Татарстан	г. Нижнекамск, ул. Заводская, 4

Таблица отгрузки товара

Номер накладной	ID дилера	Артикул	Отгружено упаковок	Дата отгрузки
001	101	01002	300	15/09/2010
002	104	01002	100	15/09/2010
003	105	01002	200	15/09/2010
004	101	02002	20	15/09/2010
005	104	02002	30	15/09/2010
006	104	01003	20	16/09/2010

Таблица товаров

Наименование товара	Артикул	Отдел	Количество единиц в упаковке
Фломастеры, набор 24 шт.	01001	Канцтовары	24
Бумага А4, пачка 500 листов	01002	Канцтовары	5
Скрепки металлические 1000 шт.	01003	Канцтовары	48
Розетки трехфазные	02001	Электротовары	12
Лампа накаливания 60 вт	02002	Электротовары	100
Выключатель 2-клавишный	02003	Электротовары	48

Сколько пачек бумаги было отгружено дилерам в Башкортостан 15 сентября 2010 г.?

- 1) 200 2) 300 3) 500 4) 2500

Решение:

Очевидно, таблиц отгрузки товара связана с таблицами товаров и зарегистрированных дилеров по полям Артикул и ID дилера соответственно. Пометим в таблице отгрузки товаров данные, относящиеся к дилерам из Башкортостана (ID дилера 101 и 105), товару Бумага (Артикул 01002), а также пометим интересующую нас дату:

Таблица отгрузки товара

Номер накладной	ID дилера	Артикул	Отгружено упаковок	Дата отгрузки
001	101	01002	300	15/09/2010
002	104	01002	100	15/09/2010
003	105	01002	200	15/09/2010
004	101	02002	20	15/09/2010
005	104	02002	30	15/09/2010
006	104	01003	20	16/09/2010

Строки, соответствующие номерам накладной 001 и 003, удовлетворяют всем параметрам запроса. Отгружено 300 и 200 упаковок бумаги. Из таблицы товаров выясняем, что в каждой упаковке было 5 пачек бумаги. Всего было отгружено $(300+200)*5=2500$ пачек бумаги.

Ответ № 4

Пример 2 задания с одним верным ответом

Дана база данных, содержащая сведения об оценках учеников:

№ записи	Фамилия	Имя	Математика	Физика	Информатика
1	Ипатов	Егор	4	4	5
2	Киреев	Александр	5	5	5
3	Илюхин	Сергей	5	4	4
4	Ивашкина	Татьяна	5	5	4

Запросу, содержащему условие отбора

$(\text{Физика} = 5 \text{ ИЛИ Информатика} = 5) \text{ И Математика} = 5$

отвечают записи

1) 1,3,4 2) 4 3) 3,4 4) 2,4

Решение: В первую очередь в логическом выражении (условии отбора) выполняется логическая операция ИЛИ, т.к. она находится в круглых скобках. Условию «Физика = 5 ИЛИ Информатика = 5» отвечают записи 1, 2 и 4. Условию «Математика = 5» отвечают записи 2, 3, 4. Для того чтобы составное логическое выражение было истинным, необходимо истинность обоих его частей. Следовательно, условиям запроса отвечают записи 2 и 4.

Покажем решение в табличном виде (поля **Фамилия** и **Имя** не показаны)

№ записи	Математика	Физика	Информатика	Физика = 5	Информатика = 5	Физика = 5 или Информатика = 5	Математика = 5	(Физика = 5 или Информатика = 5) и Математика = 5
1	4	4	5		да	да		
2	5	5	5	да	да	да	да	да
3	5	4	4					
4	5	5	4	да		да	да	да

Ответ: № 4

Решения заданий демоварианта 2012

Задание А6

Характеристики задания

Проверяемые элементы содержания	Знание технологии хранения, поиска и сортировки информации в базах данных
Контролируемый элемент содержания (по кодификатору)	3.5.1 Системы управления базами данных . Организация баз данных
Требования к уровню подготовки (по кодификатору)	2.1.2 Создавать и использовать структуры хранения данных
Вид деятельности	Применение знаний и умений в стандартной ситуации
Уровень	базовый
Максимальный первичный балл	1
Время выполнения	3 мин.

Задание

А6 В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы бабушки Ивановой А.И.

Таблица 1

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
	...	

Таблица 2

ID_Родителя	ID_Ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

- 1) Иванов Т.М.
- 2) Черных И.А.
- 3) Цейс Т.Н.
- 4) Петренко Н.Н.

Решение

- 1) По Таблице 1 определим ID Ивановой А.И. – 23.
- 2) В Таблице 2 найдем строки, в которых ID_Ребенка = 23 и определим соответствующие ID_Родителя – это значения 13 и 85.

ID_Родителя	ID_Ребенка
13	23
85	23

- 3) Затем найдем в Таблице 2 строки с ID_Ребенка 13 и 85 и определим ID_Родителя. Для ID_ребенка 13 это значения 82 и 95. Для ID_ребенка 85 в таблице 2 не приводятся данные о родителях

ID_Родителя	ID_Ребенка
82	13
95	13

- 4) В Таблице 1 значению ID = 82 соответствует мужчина Черных А.Н., это дедушка Ивановой А.И. Значению ID = 95 соответствует женщина Цейс Т.Н. – бабушка Ивановой А.И.

ID	Фамилия_И.О.	Пол
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж

Ответ: № 3

Задания для самостоятельного решения

Задание 1 с одним верным ответом

Для фрагмента базы данных «Детали»

Номер детали	Деталь	Вес	Материал
25	Гайка	13	Сталь
27	Гайка	18	Латунь
41	Шайба	17	Сталь
52	Болт	20	Чугун
53	Шайба	20	Алюминий

Условию отбора

(Деталь <> "Гайка" ИЛИ Деталь= "Шайба") И Вес <20

отвечают записи с номерами деталей

- 1) только 41 2) только 25,27 3) только 41,53 4) 25, 27, 41, 53

Задание 2 с одним верным ответом

Представлена база данных «Классы школы»

Класс	Кол_учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский

После сортировки в порядке возрастания по полю **Класс** сведения о 10б классе переместятся на

- 1) 4 строки вверх 2) 1 строку вверх 3) 3 строки вверх 4) 2 строки вверх

Задание 3 с одним верным ответом

Представлена база данных "Продажи автомобилей"

Модель	Цена	Продано
BMW	30	5
MERSEDES500	27	8
VAZ21099	10	12
Ford	22	2
UAZ	6	3

После проведения сортировки сведения об автомобиле MERSEDES500 переместились на одну строку вниз. Сортировка проводилась

- 1) В порядке возрастания по полю **Цена**
- 2) В порядке убывания по полю **Продано**
- 3) В порядке возрастания по полю **Модель**
- 4) В порядке убывания по полю **Цена**

Задание 4 с одним верным ответом

Для базы данных из примера 1 определите, сколько единиц канцтоваров было отгружено в Чувашию 15 и 16 сентября 2010 г.

- 1) 120 2) 500 3) 960 4) 1460

Задание 5 с кратким ответом

Дан фрагмент базы данных «Детали»

Номер детали	Деталь	Вес	Материал
25	Гайка	12	Сталь
27	Шуруп	18	Латунь
41	Шайба	17	Сталь
52	Болт	25	Чугун
53	Шайба	20	Чугун

Условию отбора

Материал = "Чугун" ИЛИ Материал = "Сталь" И Вес <20

в заданном фрагменте соответствует ____ записей. В ответе запишите одно число.

Задание 6 с кратким ответом

Дан фрагмент базы данных «Детали»

Номер строки	Номер детали	Деталь	Вес	Материал
1	25	Гайка	12	Сталь
2	27	Шуруп	18	Латунь
3	41	Шайба	17	Сталь
4	52	Болт	20	Чугун
5	53	Шайба	20	Алюминий

Условию отбора

(Материал = "Чугун" ИЛИ Материал = "Сталь") И Вес >= 15

в заданном фрагменте соответствует ____ записей. В ответе запишите одно число.