Глава 3 БАЗЫ ДАННЫХ. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД)

При изучении данной главы рекомендуется установить следующее программное обеспечение:

для операционной системы Windows:

 программу составления генеалогических деревьев GenoPro;

для операционных систем Windows и Linux:

• систему управления базами данных (СУБД) OpenOffice Base.

3.1. Базы данных

Для упорядоченного хранения и обработки связанных между собой данных используются базы данных.

База данных (БД) представляет собой определённым образом организованную совокупность данных некоторой предметной области, хранящуюся в компьютере и постоянно используемую.

Библиотечный каталог — пример базы данных. Он хранит информацию о книгах, каждая из которых имеет название, автора, год издания и т. д.

Информация в базах данных хранится в упорядоченном виде. Так, в библиотечном каталоге все записи упорядочены либо по алфавиту (алфавитный каталог), либо по области знания (предметный каталог).

База данных является информационной моделью организации данных предметной области.

Базы данных можно классифицировать по различным признакам. Рассмотрим их классификацию по используемой модели

www



данных. Принято выделять реляционные, иерархические и сетевые модели данных. В настоящее время реляционные БД являются доминирующими, поэтому о них в основном и пойдёт речь в учебнике.

Реляционные базы данных. Реляционная модель базы данных была предложена в 1969 г. математиком, научным сотрудником фирмы IBM Э. Ф. Коддом.

Слово «реляционная» происходит от английского *relation* — отношение. Это строгое математическое понятие, относящееся к теории множеств. Для пользователя базы данных отношения удобно представлять в виде неупорядоченных таблиц. Таблицы состоят из столбцов и строк и содержат данные.

В виде двумерной таблицы удобно предствалять базу данных, хранящую данные о группе объектов с одинаковыми свойствами. В каждой строке таблицы последовательно размещаются значения свойств одного из объектов; каждое значение свойства — в своём столбце, озаглавленном именем свойства.

Столбцы такой таблицы называют полями; каждое поле характеризуется своим именем (именем соответствующего свойства) и типом данных, представляющих значения данного свойства.

Поле базы данных — это столбец таблицы, содержащий значения определённого свойства.

Строки таблицы являются записями об объекте; эти записи разбиты на поля столбцами таблицы, поэтому каждая запись представляет набор значений, содержащихся в полях.

Запись базы данных — это строка таблицы, содержащая набор значений свойств, размещённый в полях базы данных.

Каждая таблица должна содержать, по крайней мере, одно ключевое поле, содержимое которого уникально для каждой записи в этой таблице. Ключевое поле позволяет однозначно идентифицировать запись в таблице.

Ключевое поле — это поле, значения которого однозначно определяют запись в таблице.

В качестве ключевого поля чаще всего используют поле, содержащее порядковый номер. Однако иногда удобнее в качестве ключевого поля таблицы использовать другие поля, код объекта, например инвентарный номер, и т. п.

Тип поля. Тип поля определяется типом данных, которые оно содержит. В OpenOffice Base поля могут содержать данные следующих основных типов:

- Текст содержит текст длиной до 255 символов;
- Число содержит числа;
- Целое содержит целые числа;
- Вещественное содержит вещественные числа;
- Дата/Время содержит информацию о дате и времени;
- Логическое содержит значения TRUE (истина) или FALSE (ложь).

Каждый тип поля имеет свой набор свойств. Наиболее важными свойствами полей являются:

- Длина определяет максимальную длину (текстовые и числовые поля);
- *Пример формата* устанавливает формат представления данных;
- *Обязательное* указывает, что данное поле должно быть заполнено.

База данных «Процессоры». Рассмотрим, например, базу данных «Процессоры» (табл. 3.1), которая содержит перечень объектов (процессоров), каждый из которых имеет имя (название). В качестве характеристик (свойств) можно рассмотреть количество элементов в процессоре и частоту системной шины. Поле Процессор является текстовым, а поля *Технологический процесс*, Частота системной шины и, естественно, *Счётчик* — числовыми полями.

При этом каждое поле обладает определённым набором свойств. Например, для поля *Счётчик* задан формат данных *целое число*.

Таблица 3.1

Счёт- чик	Процессор	Технологический процесс, нм	Частота систем- ной шины, МГц
1	Intel Pentium III	250	100
2	Intel Core 2	65	533
3	Celeron M	90	533
4	Intel Core i7	32	2400

Табличная база данных «Процессоры»

Вопросы и задания

- 1. В чём заключается разница между записью и полем в базе данных?
- 2. Поля каких типов могут присутствовать в базе данных?
- 3. Чем отличается ключевое поле от остальных полей?

3.2. Система управления базами данных

3.2.1. Основные объекты СУБД: таблицы, формы, запросы, отчёты

Системы управления базами данных (СУБД). Развитие информационных технологий привело к созданию компьютерных баз данных. Создание баз данных, а также операции поиска и сортировки данных выполняются специальными программами системами управления базами данных (СУБД). Таким образом, необходимо различать собственно базы данных (БД), которые являются упорядоченными наборами данных, и системы управления базами данных (СУБД) — программы, управляющие хранением и обработкой данных.

Система управления базами данных (СУБД) — это программа, позволяющая создавать базы данных, а также обеспечивающая обработку (сортировку) и поиск данных.

В реляционных СУБД Microsoft Access и OpenOffice Base используется стандартный для операционных систем многооконный интерфейс, но, в отличие от других приложений, не многодокументный. Единовременно может быть открыта только одна база данных, содержащая обязательное окно базы данных и окна для работы с объектами базы данных. В каждый момент времени одно из окон является активным, и в нём курсором отмечается активный объект.

Окно базы данных — один из главных элементов интерфейса СУБД. Здесь систематизированы все объекты базы данных: таблицы, запросы, формы, отчеты.

Таблицы. В реляционных базах данных вся информация хранится в двумерных таблицах. Это *базовый* объект базы данных, все остальные объекты создаются на основе существующих таблиц (производные объекты). Каждая строка в таблице — *запись* базы данных, а столбец — *поле*. Запись содержит набор данных об одном объекте, а поле — однородные данные обо всех объектах.

Запросы. В СУБД запросы являются важнейшим инструментом. Главное предназначение запросов — это отбор данных на основании заданных условий.

Формы. Формы позволяют отображать данные, содержащиеся только в одной записи. При помощи форм можно добавлять в таблицы новые данные, а также редактировать или удалять существующие. Форма может содержать рисунки, графики и другие внедрённые объекты.

Отчёты. Они предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде.

Существует достаточно много различных СУБД, но для первого знакомства мы рекомендуем СУБД OpenOffice Base.

Вопросы и задания

- 1. Какую функцию выполняют СУБД?
- 2. Можно ли открыть в СУБД несколько разных баз данных?
- Какие основные объекты входят в СУБД и какие функции они выполняют?

Практическая работа 3.1

Создание базы данных

Задание. В СУБД OpenOffice Base создать базу данных «Процессоры» (см. табл. 3.1).

Варианты выполнения работы:

- включить в базу данных различные поля;
- заполнить базу данных различными записями.



Создание базы данных «Процессоры»



Электронное приложение к главе 3.

- 1. В операционной системе Windows или в операционной системе Linux запустить СУБД OpenOffice Base.
- 2. В появившемся диалоговом окне *Macmep баз данных* (шаг 1) выбрать с помощью переключателя пункт *Создать новую базу данных* (рис. 3.1).

Щёлкнуть по кнопке Далее >>.

🗩 Мастер баз данных				
Шаги	Добро пожаловать в Мастер баз данных OpenOffice			
1. Выбор базы данных 2. Сохранить и выполнить	Вы должны создать новую базу или настроить соединение к существующей базе данных. В обоих случаях будет создан файл, содержащий настройки соединения, формы и отчёты.			
	Что вы хотите сделать?			
	Осоздать новую базу данных			
	Открыть существующий <u>ф</u> айл			
	Предыдущие New Database			
	🖉 Открыт <u>ь</u>			
	Подключиться к существующей базе данных			
	JDBC <			
<u>С</u> правка	<< <u>Н</u> азад Далее >> <u>Г</u> отово Отмена			

Рис. 3.1

3. В появившемся диалоговом окне *Мастер баз данных* (шаг 2) установить флажок *Открыть базу для редактирования* (рис. 3.2).

Щёлкнуть по кнопке Готово.

- 4. Откроется диалоговое окно Сохранить как, в котором выбрать место сохранения и имя базы данных («Процессоры»).
- 5. В левой части появившегося диалогового окна Процессоры в разделе База данных выбрать пункт Таблицы, в средней части окна в разделе Задачи выбрать пункт Создать таблицу в режиме дизайна... (рис. 3.3).
- 6. В появившемся окне Таблица 1 в столбце Название поля ввести имена полей базы данных «Процессоры». В столбце Тип поля с помощью раскрывающегося списка выбрать для каждого поля его тип (рис. 3.4).
- 7. Сохранить структуру базы данных [Файл—Сохранить]: в диалоговом окне Сохранить как ввести Имя таблицы — «Таблица 1»; нажать кнопку ОК.
- 8. Закрыть окно Таблица1—OpenOffice.org Base: Table Design.



Рис. 3.2



Рис. 3.3

😌 Процессоры.odb : Таблица1	- OpenOffice Base: Table Desi	ign □ Σ	3	
<u>Ф</u> айл <u>П</u> равка <u>В</u> ид С <u>е</u> рвис	<u>О</u> кно <u>С</u> правка	&		
	C .			
Название поля	Тип поля	Описание		
ID	Целое [INTEGER]		^	
Процессор	Tekct [VARCHAR]		Ε	
Технологический процесс	Десятичное [DECIMAL]		H	
Image: Construction of the second o	Текст (фикс.) [CHAR] Число (NUMERIC.) Десятичное [DecIMAL] Целое [INTEGER] Короткое целое [SMALLIN С плавающей точкой [FAL] С дойхий точностью [O C Текст [VARCHAR] Текст [VARCHAR] Логическое [BOOLEAN] Дата [DATE] Дата Драте] Дата / Время [TIMESTAMP ПРО/ПЕ [OTHER]	Image: Trip in the second s		
<u>، (</u>	1	m	ĺ	
		Свойства поля		
Обязательное Длина Знаков после запятой	Her 50		* III	
Значение по умолчанию				
Пример формата	0		Ŧ	

Рис. 3.4

9. В диалоговом окне Процессоры в разделе Таблицы появится пункт Таблица 1.

Ввод данных в таблицу базы данных и их редактирование мало чем отличаются от аналогичных действий в других офисных приложениях.

При вводе данных в режиме *Таблица* в поле маркера записи, которое расположено слева от полей таблицы, может отображаться один из следующих символов:



(звёздочка) — обозначает пустую запись в конце таблицы;

(стрелка) — обозначает выделенную (активную) запись;

(карандаш) — обозначает, что в записи производятся изменения.

10. Осуществить двойной щелчок по пункту Таблица 1 и в появившемся диалоговом окне Таблица 1—Процессоры— OpenOffice Base: Table Data View ввести записи базы данных «Процессоры» (рис. 3.5).

😌 Таблица1 - Процессоры - OpenOffice Base: Table Data View 🗖 🔲 🗵					23		
₫ai	йл <u>П</u> р	равка <u>В</u> ид Вс	т <u>а</u> вка С <u>е</u> рвис <u>О</u> кно <u>С</u> правка		&		
	ID	Процессор	Технологический процесс нм	Частота системной шины МГц			
	1	Intel Pentium II	250	100			
	2	Intel Core 2	65	533			
	3	Celeron M	90	533			
	4	Intel Core i7	32	2400			
ø							
Запі	ись 1	ИЗ	4				

Рис. 3.5

Перемещение между записями можно осуществлять с помощью мыши, клавиш управления курсором или полосы прокрутки. Для быстрого перемещения между записями в базе данных можно использовать кнопки перемещения на панели *Запись*, которая находится в нижней части окна таблицы.

3.2.2. Использование формы для просмотра и редактирования записей в базе данных

Записи базы данных можно просматривать и редактировать в виде **таблицы** или в виде **формы**. Выше мы работали с базой данных, представленной в виде таблицы, когда запись образует строку в этой таблице. Такое представление базы данных позволяет наблюдать несколько записей одновременно, и в этом состоит достоинство табличного представления.

Однако часто вид *Таблица* не позволяет видеть полностью всю информацию на экране. Если база данных содержит достаточно много полей, а значения полей содержат много символов, то не все поля таблицы могут умещаться на экране, а значения полей могут быть видны не полностью.

 $\Phi opma$ отображает одну запись в удобном для пользователя виде. В процессе создания формы можно указать, какие поля базы данных включить в форму, как расположить поля в окне формы, а также как можно сделать форму визуально привлекательной.

Фактически с помощью формы создаётся графический интерфейс доступа к базе данных, который может содержать различные элементы управления (текстовые поля, кнопки, переключатели и т. д.), а также надписи. Обычно на форме размещаются надписи, являющиеся именами полей базы данных, и текстовые поля, содержащие данные из базы данных.

Пользователь может изменять дизайн формы (размер, цвет и т. д.), элементов управления и надписей.

Примерами форм могут являться *Визитка* в базе данных «Записная книжка» или *Карточка* в базе данных «Библиотечный каталог», которые содержат лишь одну запись базы данных, зато представленную в удобном для пользователя виде.

Вопросы и задания

В чём заключается разница между представлением базы данных с помощью таблицы и формы?

Практическая работа 3.2

Создание формы в базе данных

Задание. В СУБД OpenOffice Base создать форму для базы данных «Процессоры».

- Варианты выполнения работы:
- задать различный дизайн формы и расположение полей и надписей на форме.



Создание формы для базы данных «Процессоры»



Электронное приложение к главе 3.

Создание формы можно проводить различными способами:

- самостоятельно (в *Режиме дизайна*) этот способ позволяет начать создание формы «с нуля»;
- с использованием *Macmepa форм* (более простой путь), который с помощью серии диалоговых окон помогает пользователю в создании формы.

Создадим форму для базы данных «Процессоры» с помощью Мастера форм.

1. В левой части появившегося диалогового окна Процессоры в разделе База данных выбрать пункт Формы, в средней части окна в разделе Задачи выбрать пункт Использовать мастер для создания формы ...

2. В появившемся диалоговом окне *Мастер форм* (шаг 1: *Выбор поля*) выбрать поля базы данных из *Таблицы* 1, которые будут отображаться на форме (рис. 3.6). Щёлкнуть по кнопке *Дальше* >.

Мастер форм	E C
Шаги	Выберите поля формы
 Выбор поля Установка субформы Добавить поля субформы Получить объединённые поля Расположить элементы управления Установка источника данных Применить стили Задать имя 	Таблица: Таблица1 Существующие поля Существующие поля Существующие поля Поля в форме Процессор Технологический процесс Частота системной шины С
<u>С</u> правка	< Назад Дальше > Готово Отмена

Рис. 3.6

- 3. В диалоговом окне *Мастер форм* (шаг 5: *Расположить элементы управления*) выбрать расположение полей, которые будут отображаться на форме (рис. 3.7). Щёлкнуть по кнопке *Дальше* >.
- 4. В диалоговом окне Мастер форм (шаг 6: Установка источника данных) установить переключатель в положение Форма для отображения всех данных.
 Шёлкнуть по кнопке Дальше >.
- 5. В диалоговом окне *Мастер форм* (шаг 7: *Применить стили*) выбрать стиль и обрамление поля (рис. 3.8).

Щёлкнуть по кнопке Дальше >.

6. В диалоговом окне *Мастер форм* (шаг 8: Задать имя) ввести название формы (например, Форма 1) и установить переключатель в положение *Модифицировать форму*. Щёлкнуть по кнопке Готово.

Мастер форм	X
Шаги	Расположите элементы управления на Вашей форме
 Выбор поля Установка субформы Добавить поля субформы Получить объединённые поля 	Расположение подписи По левому краю По правому краю Расположение головной формы
 Б. Расположить элементы управления Установка источника данных 	Столбцы - подписи слева
7. Применить стили 8. Задать имя	Расположение субформ
<u>С</u> правка	Котово

Рис. 3.7

Мастер форм		
 Шаги	Применить стиль к форме	
 Выбор поля Установка субформы Добавить поля субформы Получить объединённые поля Расположить элементы управления Установка источника данных Применить стили Задать имя 	Прим <u>е</u> нить стили Бежевый Фиолетовый Светло-синий Светло-серый Тёмный Оранжевый Сине-голубой Серый Вода Красный	Обрамление поля © Без обрамлени <u>я</u> @ Трёхмерны <u>й</u> вид © Плоские
<u>С</u> правка	< Назад Дальше >	Готово Отмена

Рис. 3.8

 Получим заготовку формы, в которой слишком мелкий шрифт, размеры некоторых полей не вмещают данные и т. д. (рис. 3.9).
 Закройте форму.

🖹 Про	🖹 Процессоры.odb : Таблица1 - OpenOffice Base: Database Form								
<u>Ф</u> айл	<u>П</u> равка <u>В</u> ид Вста	івка Формат	<u>Т</u> аблица	С <u>е</u> рвис	<u>О</u> кно	<u>С</u> правка			
: 🖻 ·	· 🖻 • 🔒 🗠 🛛		S ABS /	15C 🔀	fa (fa	15) • (Cl	- G	• == •
<u>Re</u>	ID								
	Процессор								
	Технологиче	кий процесс							
ABC	Частота систе	мной шины							
0									
•									
i iti	Запись 1	из 1			· 🔚	19 ⁽ x	Z	2 {	2 ∆↓
	le p. e. le le	III 🗞 🙋	B }	ů •			+ .		/ =
Стран	ица 1/1	Обычный				CTAH			

Рис. 3.9

Вид формы можно изменять в *Режиме разработки*. В созданной нами форме базы данных «Процессоры» не все надписи хорошо видны, поэтому размер шрифта нужно увеличить. Некоторые поля вывода числовых значений также нужно увеличить, чтобы числовые данные в них помещались.

- 8. Для перехода в режим разработки в главном окне базы данных вызвать контекстное меню для созданной Формы 1 и выполнить команду Изменить.
- 9. В режиме разработки выделить поле с надписью и осуществить двойной щелчок. В появившемся диалоговом окне *Свойства* установить необходимые параметры (в том числе размер шрифта). Проделать это для всех полей (рис. 3.10).

Свойства: Групповое выделение		B
Общие		
Имя		
Доступно	Да 💌	
Видимость	Да 💌	
Для печати	Да 💌	
Привязка	К абзацу 💌	
Позиция Х	▲ ▼	
Позиция Ү	▲ ▼	
Ширина	▲ ▼	
Высота	▲ ▼	
Шрифт	(По умолчанию)	
Выравнивание	По левому краю 🔻	
Верт. выравнивание	По умолчанию 💌	
Цвет фона	-	
Обрамление	-	
Цвет рамки	По умолчанию 🔻	
Дополнительная информация		
Текст всплывающей подсказки		
URL Справки		

Рис. 3.10

Для изменения параметров надписи и поля по отдельности их нужно разгруппировать.

10. Выделить надпись и поле, в контекстном меню выбрать пункт [Группировка—Разгруппировать]. Проделать это для всех полей.

Теперь форма для базы данных «Процессоры» готова. С её помощью можно просматривать записи, редактировать их или вводить новые записи.

После открытия форма содержит запись \mathbb{N} 1. При работе с формой для перехода от одной записи к другой необходимо воспользоваться панелью *Запись*, которая находится в нижней части окна формы.

11. В окне Форма 1 щёлкнуть по кнопке Режим разработки.

С использованием панели *Запись*, которая содержит кнопки со стрелками, можно перемещаться по записям.

Также можно в поле номера записи ввести номер искомой записи (рис. 3.11).

🖹 Процессоры.odb : Таблица12 - OpenOffice Base: Database Form					
<u>Ф</u> айл <u>П</u> равка <u>В</u> ид Вст <u>а</u> вка Фо <u>р</u> мат <u>Т</u> аблица С <u>е</u> рвис <u>О</u> кно <u>С</u> правка					
Ê · P · G · B B B B · Ø ₩ ₩ ₩ ₩ ₩ ₩ 9 · C · 6 ■ •					
Базовый • Тітеs New Roman • 12 • ж К Ц 🗐 🗄 🚍					
$ \sum_{i=1}^{n} \cdots 1_{i_1} \cdots 2_{i_1} \cdots 3_{i_1} \cdots 4_{i_1} \cdots 5_{i_1} \cdots 6_{i_1} \cdots 7_{i_1} \cdots 8_{i_1} \cdots 9_{i_1} \cdots 11_{i_1} \cdots 12_{i_1} \cdots 13_{i_1} \cdots 13_{i_n} \cdots 11_{i_n} \cdots 11_$					
• ID 4					
Процессор					
авс I ехнологический процесс нм 32					
Частота системной шины МГц 2400					
9 -					
& ▶ @ @ ≥ ⊞ & ▶ & + ₩ ¶ 					
Страница 1/1 Обычный Русский ВСТ СТАНД *					

Рис. 3.11

3.2.3. Поиск записей в базе данных с помощью фильтров и запросов

Поиск записей с помощью фильтров. Фильтры позволяют отбирать записи, которые удовлетворяют заданным условиям. Условия отбора записей создаются с использованием операторов сравнения (=, >, < и т. д.).

Простой фильтр содержит условие отбора записей только для одного поля. Сложный фильтр содержит несколько условий для различных полей. В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют всем условиям одновременно.

Поиск записей с помощью запросов. Запросы осуществляют поиск записей в табличной базе данных так же, как и фильт-

ры. Различие между ними состоит в том, что запрос является самостоятельным объектом базы данных, а фильтр привязан к конкретной таблице.

Запрос является производным объектом от таблицы. Однако результатом выполнения запроса является также таблица, т. е. запросы могут использоваться вместо таблиц. Например, форма может быть создана как для таблицы, так и для запроса.

Запросы позволяют отобрать те записи, которые удовлетворяют заданным условиям. Запросы, как и фильтры, бывают простые и сложные. Простой запрос содержит одно условие, а сложный запрос содержит несколько условий для различных полей.

В процессе создания запроса можно отбирать не только записи, но и поля, которые будут присутствовать в запросе.

Алгебра логики. Создание (формулирование) запросов тесно связано с понятиями алгебры логики.

Методы алгебры логики помогают нам формализовать логические рассуждения и с их помощью реализовать компьютерную обработку и поиск информации. Подобно тому как в обычной алгебре используются переменные и математические операции, алгебра логики позволяет формулировать логические выражения в абстрактной форме, используя логические переменные и выполняемые над этими переменными логические операции.

Исходными объектами в алгебре логики являются высказывания. Высказывание представляет собой повествовательное предложение, которое содержит некоторое утверждение о рассматриваемом объекте и может быть истинным или ложным.

Примеры высказываний:

- 1) «Число 2 чётное» истинное высказывание;
- 2) «Марс спутник Земли» ложное высказывание.

Вопросительные и восклицательные предложения, а также фразы, в которых содержится только название объекта без каких-либо утверждений о нём (например: «белая ворона»), не являются высказываниями.

Высказывание можно обозначить логической переменной. В зависимости от того, является ли высказывание истинным или ложным, соответствующая ему логическая переменная тоже является истинной или ложной. Истинность в алгебре логики обычно обозначают словом ИСТИНА, словом *true* или числовым значением 1. Ложность обозначают словом ЛОЖЬ, словом *false* или числовым значением 0.

Само высказывание также может содержать в себе одну или несколько логических переменных, значения которых определяют истинность или ложность высказывания. Такое высказывание,

содержащее логические переменные, называют высказывательной формой.

Высказывание может быть простым (содержит одно утверждение об одном объекте) или сложным, составленным из нескольких простых высказываний (в том числе обозначенных логическими переменными). В простых высказываниях чаще всего используются операции сравнения: «равно», «не равно», «меньше», «больше», «меньше или равно», «больше или равно». Если операции сравнения содержат математические переменные, то истинность или ложность такого простого высказывания зависит от значений этих переменных. При поиске записей в табличной базе данных простые фильтры и простые запросы формулируются в виде простых высказываний.

Сложное высказывание формулируется из простых при помощи логических операций, которые позволяют по истинности или ложности связываемых ими простых высказываний (операндов) однозначно определить истинность или ложность результата. Каждая логическая операция описывается таблицей истинности, содержащей все возможные комбинации значений операндов и соответствующие им значения результата выполнения логической операции.

Пример:

Сложное высказывание «A И B» («X > 1» И «X < 3») образовано соответствующими операндами и связано логической операцией И.

Рассмотрим основные логические операции.

НЕ (отрицание, инверсия). Обозначается символом – или чертой над логической переменной (надчёркиванием).

Логическая операция НЕ имеет только один операнд и меняет его значение с истинного на ложное и наоборот.

Таблица истинности операции НЕ:

A	\overline{A}
0	1
1	0

И (логическое умножение, конъюнкция). Обозначается символом </br>или &.

Результат логической операции И является истинным только в том случае, когда оба операнда являются истинными.

Таблица истинности операции И:

A	В	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

ИЛИ (логическое сложение, дизъюнкция). Обозначается символом v, + или |.

Результат логической операции ИЛИ является истинным, если хотя бы один операнд является истинным.

Таблица истинности операции ИЛИ:

A	В	$A \lor B$
0	0	0
0	1	1
1	0	1
1	1	1

Исключающее ИЛИ (неравнозначность). Обозначается символом \oplus .

Результат логической операции исключающее ИЛИ является истинным, только если её операнды имеют различные значения истинности.

Таблица истинности операции исключающее ИЛИ:

A	В	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Операция исключающее ИЛИ может быть заменена сочетанием операций отрицания, конъюнкции и дизъюнкции: $A \oplus B = (A \land \overline{B}) \lor (\overline{A} \land B).$

Результат импликации является ложным только тогда, когда из истины следует ложь. Во всех остальных случаях результат импликации истинен.

Таблица истинности операции импликации:

A	В	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Операция импликации может быть заменена сочетанием операций отрицания и дизъюнкции: $A \to B = \overline{A} \lor B$.

ТОГДА И ТОЛЬКО ТОГДА (эквиваленция, тождество). Обозначается символом \leftrightarrow или ~.

Результат эквиваленции является истинным, только если оба её операнда имеют равные значения (оба равны 0 или оба равны 1).

Таблица истинности операции эквиваленции:

A	В	$A \sim B$
0	0	1
0	1	0
1	0	0
1	1	1

Операция эквиваленции может быть заменена сочетанием операций отрицания, конъюнкции и дизъюнкции: $A \sim B = (\overline{A} \lor B) \land (A \lor \overline{B})$ или $A \sim B = (\overline{A} \land \overline{B}) \lor (A \land B)$.

Порядок выполнения логических операций в сложном высказывании определяется их приоритетом. Наивысший приоритет имеет операция НЕ. Второй по порядку выполняется операция И. После операций И выполняется операция ИЛИ. После дизъюнкции выполняются операции исключающее ИЛИ и импликация. Самый низкий приоритет имеет операция эквиваленции. Порядок выполнения логических операций может быть изменён при помощи скобок.

Сложные логические выражения можно преобразовать так, чтобы упростить запись (сократить количество выполняемых

логических операций) или заменить одни логические операции (исключающее ИЛИ, импликацию, эквиваленцию) на другие (операции НЕ, И, ИЛИ). Такие преобразования выполняются согласно следующим законам алгебры логики.

1. Законы ассоциативности и коммутативности: $A \land (B \land C) = (A \land B) \land C, A \lor (B \lor C) = (A \lor B) \lor C.$ 2. Законы дистрибутивности: $A \land (B \lor C) = (A \land B) \lor (A \land C), A \lor (B \land C) = (A \lor B) \land (A \lor C).$ 3. Законы идемпотентности: $A \land A = A, A \lor A = A.$ 4. Закон двойного отрицания: $\overline{\overline{A}} = A.$ 5. Закон непротиворечия: $A \land \overline{\overline{A}} = 0.$ 6. Закон исключённого третьего: $A \lor \overline{\overline{A}} = 1.$ 7. Свойства логических констант: $\overline{\overline{1}} = 0, A \leftrightarrow 1 = A, A \lor 1 = 1, \overline{\overline{0}} = 1, A \leftrightarrow 0 = 0, A \lor 0 = A$

 $\overline{1} = 0, \ A \wedge 1 = A, \ A \vee 1 = 1, \ \overline{0} = 1, \ A \wedge 0 = 0, \ A \vee 0 = A.$

8. Законы де Моргана:

 $\overline{A \wedge B} = \overline{A} \vee \overline{B}, \ \overline{A \vee B} = \overline{A} \wedge \overline{B}.$

Для сложного высказывания тоже можно составить таблицу истинности. Такая таблица должна содержать все возможные комбинации исходных значений используемых в выражении переменных и для каждой такой комбинации значений переменных — все результаты выполнения имеющихся логических операций и результат всего выражения.

Пример: логическое выражение $(A \land \overline{B}) \lor (\overline{A} \land B)$. Его таблица истинности:

A	В	\overline{A}	\overline{B}	$A \wedge \overline{B}$	$\overline{A} \wedge B$	$(A \wedge \overline{B}) \vee (\overline{A} \wedge B)$
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

При поиске записей в табличной базе данных сложные фильтры и сложные запросы формулируются в виде сложных высказываний с использованем рассмотренных выше логических операций. Кроме логических операций в алгебре логики также существуют кванторы — особые операции, определяющие степень общности высказывания.

1. Квантор общности («для всех»). Обозначается символом \forall .

Пример записи выражения: « $\forall x \ x \cdot 0 = 0$ » означает «Для всех x умножение на нуль дает в результате нуль».

2. Квантор существования («существует»). Обозначается символом ∃.

Пример записи выражения: «∃*x x* — чётное» означает «Существуют значения *x*, которые являются чётными».

3. Квантор единственности («существует одно и только одно»). Обозначается символами ∃!.

Пример записи выражения: « $\exists !x \ x \cdot x = 1$ » означает «Существует только одно значение x, произведение которого само на себя равно единице».

Вопросы и задания

В чём заключается разница между поиском записей в табличной базе данных с помощью фильтров и запросов?

Практическая работа 3.3

Поиск записей в базе данных с помощью фильтров и запросов

Задание 1. В табличной базе данных «Процессоры» осуществить поиск записей с использованием простого фильтра (например, будем искать записи, удовлетворяющие условию: технологический процесс больше 32 нм).

Задание 2. В базе данных «Процессоры» осуществить поиск записей с использованием сложного запроса.

Варианты выполнения работы:

 задать различные условия поиска записей в фильтрах и запросах.



Поиск записей с использованием простого фильтра в базе данных «Процессоры»



Электронное приложение к главе 3.

1. Открыть *Таблицу* 1 базы данных «Процессоры», дважды щёлкнув по соответствующей ссылке в окне базы данных.

Появится таблица базы данных (рис. 3.12) с кнопками на Панели инструментов, позволяющими создавать и отменять фильтры (рис. 3.13).

91	Габлиц	ца1 - Процессор	ы - OpenOffice Base: Table Data Vi	ew	23
₫ai	йл ∏	равка <u>В</u> ид Вс	т <u>а</u> вка С <u>е</u> рвис <u>О</u> кно <u>С</u> правка		2
		n x fi ti		↓ 🛠 🎸 🗸 😤 📮	
	ID	Процессор	Технологический процесс нм	Частота системной шины МГц	
₽	1	Intel Pentium 🏾	250	100	
	2	Intel Core 2	65	533	
	3	Celeron M	90	533	
	4	Intel Core i7	32	2400	
\$					
Запи	ись 1	ИЗ			

Рис. 3.12



Рис. 3.13

2. Щёлкнуть по кнопке ☑, позволяющей выбрать параметры фильтра. В появившемся диалоговом окне Фильтр по умолчанию ввести условия поиска записей (Имя поля, Условие и Значение). Фильтр будет создан (рис. 3.14).

Щёлкнуть по кнопке ОК.

В появившемся окне *Процессоры: Таблица 1* будут выведены записи, удовлетворяющие условиям поиска. В данном случае это записи 2, 3 и 4.

Фильтр по умолчанию	0			23
Критерии ——— Оператор	Имя поля	Условие	Значение	ОК
	Технологический г 💌 >	•	32	Отмена
AND 👻	- нет -	-		<u>С</u> правка
AND 👻	- нет -	-]

Рис. 3.14

3. Для того чтобы отменить фильтр и возвратить все записи базы данных, щёлкнуть по кнопке 🖗 .



Поиск записей с использованием сложного запроса в базе данных «Процессоры»



Электронное приложение к главе 3.

Создание запроса можно проводить различными способами:

- самостоятельно (в *Режиме дизайна*) этот способ позволяет начать создание запроса с нуля;
- с использованием *Macmepa запросов* (более простой путь), который с помощью серии диалоговых окон помогает пользователю в создании запроса.

Создадим сложный запрос для табличной базы данных «Процессоры» с помощью *Macmepa запросов*.

- 1. В левой части диалогового окна Процессоры в разделе База данных выбрать пункт Запросы, в средней части окна в разделе Задачи выбрать пункт Использовать мастер для создания запроса ...
- 2. В появившемся диалоговом окне *Мастер запросов* (шаг 1: Выбор полей) выбрать поля базы данных из *Таблицы* 1, которые будут отображаться в запросе (рис. 3.15).

Щёлкнуть по кнопке Дальше>.

3. В диалоговом окне Мастер запросов (шаг 3: Условие поиска) выбрать поля базы данных из Таблицы 1, Условия и Значения, которые будут в запросе (рис. 3.16).

Щёлкнуть по кнопке Готово.

В появившемся окне Запрос_Таблица 1—Процессоры— OpenOffice Base : Table Data View будут выведены записи, удовлетворяющие условиям поиска. В данном случае это записи 2 и 4.

Данная таблица, полученная в результате запроса, сохраняется в базе данных «Процессоры» как самостоятельный объект.

Мастер запросов		22
Шаги	Выберите поля (столбцы) для запроса	i
 Выбор полей Порядок сортировки Условие поиска Полное или общее Группировка Условия группировки Альтернативные названия (aliases) Обзор 	Таблицы Таблица: Таблица1 ▼ Доступные поля >> < < <	Поля в запросе: Таблица1.ID Таблица1.Процессор Таблица1.Пехнологический пр Таблица1.Частота системной к
<u>С</u> правка	< Назад Дальше >	Готово Отмена

Рис. 3.15

Мастер запросов	x
Шаги	Выберите условие поиска
1. Выбор полей 2. Порядок сортировки	 Соответствие всем из следующих Соответствие дюбому из следующих
 3. Условие поиска 4. Полное или общее 5. Группировка 6. Условия группировки 	Поля Условие Значение Таблица1.Процессор больше или равно Intel
7. Альтернативные названия (aliases) 8. Обзор	Поля Условие Значение Таблица1.Технологиче меньше 250
	Поля Условие Значение равно
<u>С</u> правка	< Назад Дальще > Готово Отмена

Рис. 3.16

3.2.4. Сортировка записей в табличной базе данных

Базы данных могут содержать тысячи записей. Часто бывает необходимо их упорядочить для пользователя, т. е. расположить в определённой последовательности. Упорядочение записей называется сортировкой. *Важно*: сортировка **не** меняет физическое расположение записей в БД, она только представляет их для пользователя в нужном порядке.

Сортировка записей производится по какому-либо полю. Значения, содержащиеся в этом поле, располагаются в определённом порядке, который определяется типом поля:

- по алфавиту, если поле текстовое;
- по величине числа, если поле числовое;
- по дате, если тип поля Дата/Время и т. д.

Сортировка записей может производиться либо по возрастанию, либо по убыванию значений полей. В процессе сортировки целостность записей сохраняется, т. е. они переносятся из одного места таблицы в другое целиком.

Сортировка записей базы данных — это их упорядочение по значениям одного из полей.

Могут реализовываться **вложенные сортировки**. Они последовательно производятся по нескольким полям. После сортировки по первому полю производится сортировка по второму полю и т. д.

Вопросы и задания

Меняет ли сортировка в базе данных содержание записей?

Практическая работа 3.4

Сортировка записей в табличной базе данных

Задание. В базе данных «Процессоры» осуществить сортировку записей по разным полям (например, по полю Процессоры).

Варианты выполнения работы:

 выбрать различные поля для сортировки записей в базе данных.



Задание. Сортировка записей в табличной базе данных «Процессоры»



- 1. Открыть Таблицу 1 базы данных «Процессоры», дважды щёлкнув по соответствующей ссылке в окне базы данных. Появится таблица базы данных с кнопками на Панели инструментов (рис. 3.17), позволяющими сортировать записи по возрастанию и Рис. 3.17 убыванию.
- 2. Выделить поле базы данных, по которому будет вестись сортировка (например, Технологический процесс).

Осуществить сортировку записей по возрастанию, щёлкнув по кнопке 🐴 на Панели инструментов.

3. Результатом будет отсортированная база данных (рис. 3.18).

9	Табли	ца1 - Про	цессоры - С	penOffice E	ase: Table Data	a View			
<u>Φ</u> a	йл <u>П</u>	равка <u>В</u>	ид Вст <u>а</u> вка	С <u>е</u> рвис	<u>О</u> кно <u>С</u> правк	a		虚	
					2 · 2 20	AU 🕅 🖓 🗸	¥.		
	ID		Процессо	op	Технологи	ческий процесс, нм	Частота системн	ой шины, МГц	
	4	Intel Co	e i7		32		2400		
	3	Celeron	М		90		533		
	2	Intel Co	e 2		65		533		
	1	Intel Per	tium III		250		100		
Зап	ись 1	i	13 4	M		•		4	

Рис. 3.18

3.2.5. Печать данных с помощью отчётов

Для вывода документов на печать используются отчёты. Отчёты являются производными объектами базы данных и создаются на основе таблиц, форм и запросов.

Вопросы и задания

Для чего нужны отчёты в табличной базе данных?







Практическая работа 3.5

Создание отчёта в базе данных

Задание. В табличной базе данных «Процессоры» создать отчёт.

Варианты выполнения работы:

• задать различные дизайны отчётов.



Создание отчёта в базе данных «Процессоры»



Электронное приложение к главе 3.

Создадим отчёт для базы данных «Процессоры» с помощью *Macmepa отчётов*.

- 1. В левой части диалогового окна Процессоры в разделе База данных выбрать пункт Отчёты, в средней части окна в разделе Задачи выбрать пункт Использовать мастер для создания отчёта ...
- 2. В диалоговых окнах *Мастер отчётов* выполнить шаги 1-6 самостоятельно.

Будет получен один из вариантов отчёта (рис. 3.19).





3.3. Иерархическая модель данных

Иерархическая модель данных графически может быть представлена как перевёрнутое дерево, состоящее из объектов различных уровней. Верхний уровень (корень) занимает один объект, второй — объекты второго уровня и т. д.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении **предка** (объект, более близкий к корню) к **потомку** (объект более низкого уровня), при этом объект-предок может не иметь потомков или иметь их несколько, тогда как объект-потомок обязательно имеет только одного предка. Объекты, имеющие общего предка, называются **близнецами**.

Базой данных, основанной на иерархической модели, является Peecrp Windows, в котором хранится вся информация, необходимая для нормального функционирования компьютерной системы (данные о конфигурации компьютера и установленных драйверах, сведения об установленных программах, настройки графического интерфейса и др.). Содержание реестра автоматически обновляется при установке нового оборудования, инсталляции программ и т. д.

Вопросы и задания

- 1. В чём заключаются характерные особенности иерархических моделей данных?
- 2. Что такое распределённая база данных?

3.4. Сетевая модель данных

Сетевая модель данных является обобщением иерархической за счёт того, что на связи между объектами в сетевых моделях не накладывается никаких ограничений.

Сетевой моделью данных является, например, генеалогическое древо семьи, так как потомки (объекты нижележащего уровня) имеют всегда более одного предка (объекта вышестоящего уровня).

Программа GenoPro (Живая Родословная) предназначена для создания, редактирования, распечатывания и представления в Интернете больших генеалогических деревьев, которые являются сетевыми моделями. Особенностью программы является лёгкость, с которой можно создавать очень сложные родословные, включающие иллюстративные материалы и различные комментарии. ?

Вопросы и задания

В чём заключаются характерные особенности сетевой модели данных?

Практическая работа 3.6

Создание генеалогического древа семьи

Задание. Создать базу данных «Генеалогическое древо семьи». Варианты выполнения работы:

• построить генеалогическое древо разных семей.





Создание базы данных «Генеалогическое древо семьи»

Электронное приложение к главе 3.

Создадим основанную на сетевой модели данных базу данных «Генеалогическое древо семьи» в программе GenoPro (Живая Родословная).

- 1. Запустить программу GenoPro (Живая Родословная) командой [*Bce программы-GenoPro*].
- 2. В появившемся диалоговом окне программы на Панели инструментов выбрать себя, т. е. воспользоваться кнопками (Новый мужчина) или (Новая женщина).
- 3. С помощью кнопки 🦞 на Панели инструментов вставить в генеалогическое древо своих родителей.
- Поочерёдно выделить родителей и с помощью той же кнопки
 вставить в генеалогическое древо родителей своих родителей, т. е. дедушек и бабушек.
- 5. Поочерёдно выделить всех родственников в генеалогическом древе семьи и в контекстном меню объектов выбрать пункт *Свойства*. В появившемся диалоговом окне ввести сведения о родственнике (рис. 3.20).

В результате мы получим генеалогическое древо семьи, которое является примером сетевой модели данных (рис. 3.21).

гиперссылки	Отображение	Семья	Вкладки поль:	зователя
Общие данные Рождение итул Имя	Смерть Образование Отчество	Занятость Изобрах	Контакты Изображения кение	Источники
Рамилия/девичья Фамилия розвище/инициалы	Последняя фамилия С Альтернативные имена	/ффикс		
олное имя тображаемое имя Сын	▼ 3 ÷	строк		
Дата рождения Дата смерти Возраст Бездетность 🔍	Пол Ф Мужской С Женский С Животное С Неизвестно	Коммента	рий	
од занятий/профессия од занятий/профессия од занятий/профессия од занятий/профессия	Организация/работодатель ий телефонРабочий телефо	DH		

Рис. 3.20



Рис. 3.21

vwv

ЭОР к главе 3 на сайте ФЦИОР (http://fcior.edu.ru)

- Ввод данных в БД
- Запросы на выборку данных
- Методы и средства разработки приложений. Системы управления базами данных
- Назначение и классификация баз данных
- Поиск информации в БД. Формирование условий
- Понятие СУБД. Классификация СУБД
- Проектирование баз данных
- Проектирование объектов данных
- Проектирование отчётов
- Проектирование экранных форм
- Табличное и графическое представление данных