

Тема 6. Технология баз данных

Основные вопросы

1. Основные понятия, используемые при работе с базами данных.
2. Языки баз данных.
3. Классификация баз данных.
4. СУБД Microsoft Access.
5. Область использования.

1. Основные понятия, используемые при работе с базами данных

С самого начала развития вычислительной техники образовались два основных направления ее использования. Первое направление – применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную. Становление этого направления способствовало интенсификации методов численного решения сложных математических задач, развитию класса языков программирования, ориентированных на удобную запись численных алгоритмов, становлению обратной связи с разработчиками новых архитектур ЭВМ.

Второе направление – это использование средств вычислительной техники в автоматических или автоматизированных информационных системах. В самом широком смысле информационная система представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса. Обычно объемы информации, с которыми приходится иметь дело таким системам, достаточно велики, а сама информация имеет достаточно сложную структуру. Классическими примерами информационных систем являются банковские системы, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т.д.

Одним из естественных требований к таким системам является быстрота выполнения операций.

Эти системы главным образом ориентированы на хранение, выбор и модификацию постоянно существующей информации. Структура информации зачастую очень сложна, и хотя структуры данных различны в разных информационных системах, между ними часто бывает много общего. На начальном этапе использования вычислительной техники для управления информацией проблемы структуризации данных решались индивидуально в каждой ин-

формационной системе. Производились необходимые надстройки над файловыми системами (библиотеки программ), подобно тому, как это делается в компиляторах, редакторах и т.д.

Но поскольку информационные системы требуют сложных структур данных, эти дополнительные индивидуальные средства управления данными являлись существенной частью информационных систем и практически повторялись от одной системы к другой. Стремление выделить и обобщить общую часть информационных систем, ответственную за управление сложно структурированными данными, явилось первой побудительной причиной создания СУБД. Очень скоро стало понятно, что невозможно обойтись общей библиотекой программ, реализующей над стандартной базовой файловой системой более сложные методы хранения данных.

Рассмотрим далее основные понятия, используемые при работе с базами данных.

База данных (БД, database) – поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Предметная область – некоторая часть реально существующей системы, функционирующая как самостоятельная единица. Полная предметная область может представлять собой экономику страны или группы союзных государств, однако на практике для информационных систем наибольшее значение имеет предметная область масштаба отдельного предприятия или корпорации.

Система управления базами данных (СУБД) – комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой.

Реляционная БД – основной тип современных баз данных. Состоит из таблиц, между которыми могут существовать связи по ключевым значениям.

Таблица базы данных – регулярная структура, которая состоит из однотипных строк (записей), разбитых на столбцы (поля).

Ключевой элемент таблицы (ключ) – такое ее поле (простой ключ) или строковое выражение, образованное из значений нескольких полей (составной ключ), по которому можно определить значения других полей для одной или нескольких записей таблицы. На практике для использования ключей создаются индексы - служебная информация, содержащая упорядоченные све-

дения о ключевых значениях. В реляционной теории и концептуальной модели понятие "ключ" применяется для атрибутов отношения или сущности.

Первичный ключ – главный ключевой элемент, однозначно идентифицирующий строку в таблице. Могут также существовать альтернативный и уникальный ключи, служащие также для идентификации строк в таблице.

Связь – функциональная зависимость между объектами. В реляционных базах данных между таблицами устанавливаются связи по ключам, один из которых в главной таблице – первичный. Второй - внешний ключ - во внешней таблице. Информация о связях сохраняется в базе данных.

Целостность данных – набор правил, обеспечивающих соответствие ключевых значений в связанных таблицах.

Правило соответствия внешних ключей первичным – основное правило соблюдения условий ссылочной целостности. Для каждого значения внешнего ключа должно существовать соответствующее значение первичного ключа в родительской таблице.

Ссылочная целостность может нарушиться в результате операций вставки (добавления), обновления и удаления записей в таблицах. В определении ссылочной целостности участвуют две таблицы - родительская и дочерняя, для каждой из них возможны эти операции, поэтому изменения могут привести к нарушению ссылочной целостности.

2. Языки баз данных

Для работы с базами данных используются специальные языки, в целом называемые *языками баз данных*. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка - *язык определения схемы БД (SDL - Schema Definition Language)* и *язык манипулирования данными (DML - Data Manipulation Language)*. SDL служил главным образом для определения логической структуры БД, т.е. той структуры БД, какой она представляется пользователям. DML содержал набор операторов работы с данными, т.е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоя-

щее время реляционных СУБД является язык SQL (Structured Query Language).

Прежде всего, язык SQL сочетает средства SDL и DML, т.е. позволяет определять схему реляционной БД и манипулировать данными. При этом именование объектов БД (для реляционной БД - именование таблиц и их столбцов) поддерживается на языковом уровне в том смысле, что компилятор языка SQL производит преобразование имен объектов в их внутренние идентификаторы на основании специально поддерживаемых служебных таблиц-каталогов

Язык SQL содержит специальные средства определения ограничений целостности БД, которые хранятся в специальных таблицах-каталогах.

Авторизация доступа к объектам БД производится также на основе специального набора операторов SQL. Идея состоит в том, что для выполнения операторов SQL разного вида пользователь должен обладать различными полномочиями. Пользователь, создавший таблицу БД, обладает полным набором полномочий для работы с этой таблицей. В число этих полномочий входит полномочие на передачу всех или части полномочий другим пользователям, включая полномочие на передачу полномочий. Полномочия пользователей описываются в специальных таблицах-каталогах, контроль полномочий поддерживается на языковом уровне.

3. Классификация баз данных

По *технологии обработки* данных базы данных подразделяются на централизованные и распределенные.

Централизованная база данных хранится в памяти одной вычислительной системы и доступ к ней организуется с использованием терминалов.

Распределенная база данных состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей, которые хранятся в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По способу доступа к данным базы данных разделяются на базы данных с локальным доступом и базы данных с сетевым доступом.

Для всех современных баз данных можно организовать сетевой доступ с многопользовательским режимом работы.

По *типу модели*, лежащей в основе организации данных, базы данных подразделяются на иерархические, сетевые и реляционные.

Иерархические базы данных. В основе данной модели – иерархическая модель данных. В этой модели имеется один главный объект и остальные – подчиненные – объекты, находящиеся на разных уровнях иерархии. Взаимосвязи объектов образуют иерархическое дерево с одним корневым объектом.

Иерархическая БД состоит из упорядоченного набора нескольких экземпляров одного типа дерева. Автоматически поддерживается целостность ссылок между предками и потомками. Основное правило: никакой потомок не может существовать без своего родителя (рис. 1).

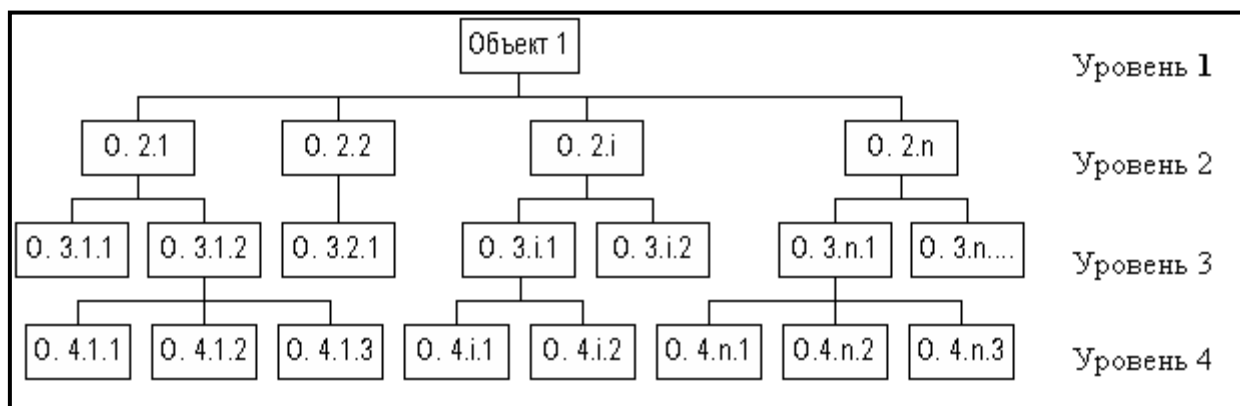


Рис 1. Схема иерархической модели данных

Типичным представителем (наиболее известным и распространенным) является Information Management System (IMS) фирмы IBM. Первая версия появилась в 1968 г. До сих пор поддерживается много баз данных этой системы.

Сетевые базы данных. Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков.

В сетевой модели данных любой объект может быть одновременно и главным, и подчиненным, и может участвовать в образовании любого числа взаимосвязей с другими объектами. Сетевая БД состоит из набора записей и набора связей между этими записями, а если говорить более точно – из набора экземпляров каждого типа из заданного в схеме БД набора типов записи и набора экземпляров каждого типа из заданного набора типов связи (рис. 2).

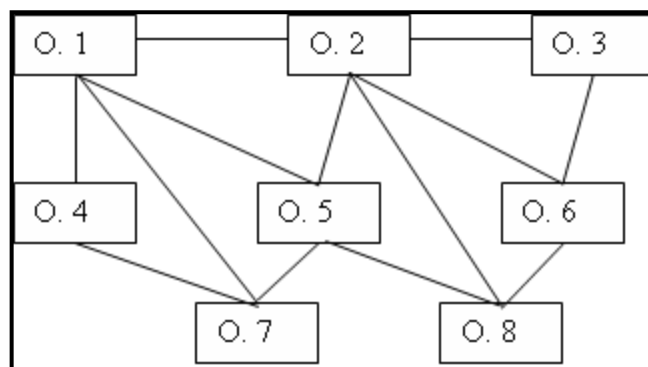


Рис. 2. Схема сетевой модели

Типичным представителем является Integrated Database Management System (IDMS) компании Cullinet Software, Inc., предназначенная для использования на машинах основного класса фирмы IBM под управлением большинства операционных систем.

Данные модели имели свои достоинства и недостатки. К достоинствам можно отнести:

- развитые средства управления данными во внешней памяти на низком уровне;
- возможность построения вручную эффективных прикладных систем;
- возможность экономии памяти за счет разделения подобъектов (в сетевых системах).

Недостатки:

- слишком сложно пользоваться;
- фактически необходимы знания о физической организации;
- прикладные системы зависят от этой организации;
- их логика перегружена деталями организации доступа к БД.

Реляционные базы данных. Реляционные системы далеко не сразу получили широкое распространение. В то время как основные теоретические результаты в этой области были получены еще в 70-х годах 20 века и тогда же появились первые прототипы реляционных СУБД, долгое время считалось невозможным добиться эффективной реализации таких систем. Однако постепенное накопление методов и алгоритмов организации реляционных баз данных и управления ими привели к тому, что уже в середине 80-х годов реляционные системы практически вытеснили с мирового рынка ранние СУБД.

Мы приступаем к изучению реляционных баз данных и систем управления реляционными базами данных. Этот подход является наиболее распространенным в настоящее время, хотя наряду с общепризнанными достоин-

ствами обладает и рядом недостатков. К числу достоинств реляционного подхода можно отнести:

- сравнительно простое моделирование большей части распространенных предметных областей и точные формальные определения;
- наличие простого и в то же время мощного математического аппарата, опирающегося главным образом на теорию множеств и математическую логику и обеспечивающего теоретический базис реляционного подхода к организации баз данных;
- возможность манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти.

В настоящее время основным предметом критики реляционных СУБД является не их недостаточная эффективность, а присущая этим системам некоторая ограниченность (прямое следствие простоты) при использовании в так называемых нетрадиционных областях (наиболее распространенными примерами являются системы автоматизации проектирования), в которых требуются предельно сложные структуры данных. Еще одним часто отмечаемым недостатком реляционных баз данных является невозможность адекватного отражения семантики предметной области. Другими словами, возможности представления знаний о семантической специфике предметной области в реляционных системах очень ограничены. Современные исследования в области постреляционных систем главным образом посвящены именно устранению этих недостатков.

Начиная с 1980-х годов, одновременно с широким распространением персональных компьютеров, большое распространение получили так называемые "настольные" реляционные СУБД, такие как dBase, FoxBase (его более поздние версии - FoxPro и Visual FoxPro), Paradox, Access. Наиболее распространенным форматом таблиц подобных реляционных баз стал *.dbf, с которым работали dBase, FoxBase, а также Clipper - система написания программ для работы с базами данных. В последующем некоторые из них стали полноценными сетевыми СУБД, имеющими возможности для работы с серверами баз данных в архитектуре "клиент-сервер", а также разработки и использования html-страниц для работы с базами данных.

Все СУБД для ПК можно подразделить на 3 вида:

- системы управления базами данных в буквальном смысле этого термина, для которых работа с базами возможна только после запуска в работу этой системы без возможности создания автономных программ, работающих с базами. К этим системам относятся: Access, Paradox, dBase.

- системы, имеющие как средства для работы с базами данных, так и возможности разработки исполняемых в операционной системе пользовательских программ (приложений), т. е. средства разработчика программ – FoxPro.

- системы для разработки пользовательских программ для работы с базами данных – Clipper, Clarion.

Все современные СУБД поддерживают режимы работы в локальной сети многих пользователей с одной базой данных. Некоторые имеют "мастеров", "построителей" и "генераторы выражений" для ускоренной разработки баз данных, экранных форм, отчетов, стандартных приложений.

В настоящее время известны также так называемые "постреляционные" СУБД, в основе которых лежат модель данных в виде многомерных таблиц (например в системе Cache фирмы InterSystems Corporation) и широкое использование принципов объектно-ориентированного подхода при организации баз данных и программировании.

4. СУБД Microsoft Access

Рассмотрим для примера СУБД Microsoft Access, входящую в число программных приложений MS Office.

Данная СУБД (система управления базами данных) является системой управления реляционной базой данных, включающей все необходимые инструментальные средства для создания локальной базы данных, общей базы данных в сети с файловым сервером или создания приложения пользователя, работающего с базой данных на SQL-сервере.

Достоинством MS Access является то, что она имеет очень простой графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать приложения, используя встроенные средства. В отличие от других настольных СУБД, MS Access хранит все данные в одном файле, хотя и распределяет их по разным таблицам, как и положено реляционной СУБД.

К этим данным относится не только информация в таблицах, но и другие объекты базы данных, которые будут описаны ниже. Для выполнения почти всех основных операций MS Access предлагает большое количество Мастеров, которые делают основную работу за пользователя при работе с данными и разработке приложений, помогают избежать рутинных действий и облегчают работу неопытному в программировании пользователю.

Создание многопользовательской базы данных и получение одновременного доступа нескольких пользователей к общей базе данных возможно в локальной сети или в сети с файловым сервером. Сеть обеспечивает аппаратную и программную поддержку обмена данными между компьютерами.

MS Access следит за разграничением доступа разных пользователей к БД и обеспечивает защиту данных. Однако возможности по обеспечению многопользовательской работы несколько ограничены. Обычно для доступа к данным по сети с нескольких рабочих станций, файл БД Access выкладывается на файловый сервер. При этом обработка данных ведется в основном на клиенте – там, где запущено приложение, в силу принципов организации файловых СУБД. Этот фактор ограничивает использование Access для обеспечения работы множества пользователей (более 15-20) и при большом количестве данных в таблицах, так как многократно возрастает нагрузка на сеть.

В плане поддержки целостности данных Access отвечает только моделям БД небольшой и средней сложности. В отношении защиты информации и разграничения доступа Access не имеет надежных стандартных средств. В стандартные способы защиты входит защита с использованием пароля БД и защита с использованием пароля пользователя. Снятие такой защиты не представляет сложности для специалиста.

Однако при известных недостатках MS Access обладает большим количеством преимуществ по сравнению с системами подобного класса. В первую очередь можно отметить распространенность, которая обусловлена тем, что Access является продуктом компании Microsoft, программное обеспечение и операционные системы которой использует большая часть пользователей персональных компьютеров. MS Access полностью совместим с операционной системой Windows, постоянно обновляется производителем, поддерживает множество языков. В целом MS Access предоставляет большое количество возможностей за сравнительно небольшую стоимость. Также необходимо отметить ориентированность на пользователя с разной профессиональной подготовкой, что выражается в наличии большого количества вспомогательных средств (Мастеров, как уже отмечалось), развитую систему справки и понятный интерфейс.

Эти средства облегчают проектирование, создание БД и выборку данных из нее. MS Access предоставляет в распоряжение непрограммирующему пользователю разнообразные диалоговые средства, которые позволяют ему создавать приложения, не прибегая к разработке запросов на языке SQL или к программированию макросов или модулей на языке VBA. MS Access обла-

дает широкими возможностями по импорту/экспорту данных в различные форматы, от таблиц MS Excel и текстовых файлов, до практически любой серверной СУБД через механизм ODBC. Еще одно немаловажное преимущество MS Access заключается в развитых встроенных средствах разработки приложений. Большинство приложений, распространяемых среди пользователей, содержит тот или иной объем кода VBA (Visual Basic for Applications).

Поскольку VBA является единственным средством для выполнения многих стандартных задач в Access (работа с переменными, построение команд SQL во время работы программы, обработка ошибок, использование Windows API и т. д.), для создания более-менее сложных приложений необходимо его знание и знание объектной модели MS Access. Одним из средств программирования в Access является язык макрокоманд. Программы, созданные на этом языке, называются макросами и позволяют легко связывать отдельные действия, реализуемые с помощью форм, запросов, отчетов. Макросы управляются событиями, которые вызываются действиями пользователями при диалоговой работе с данными через формы или системными событиями. Получается что Access, обладая всеми чертами СУБД, предоставляет и дополнительные возможности.

Это не только гибкая и простая в использовании СУБД, но и система для разработки работающих с базами данных приложений.

Функциональные возможности MS Access. Рассмотрим подробнее основные функции MS Access, чтобы иметь более ясное представление о его возможностях. В Access база данных обозначает файл, содержащий набор информации. База данных в Access может содержать следующие типы объектов: таблица, запрос, форма, отчет, страница, макрос, модуль.

Окно объектов базы данных Access может работать одновременно только с одной базой данных. Но одна БД Access может включать множество таблиц, форм, запросов, отчетов, макросов и модулей, которые хранятся в одном файле.

Таблица – это объект, соответствующий понятию «таблица» в теории реляционных баз данных. Для каждой таблицы в Access можно определить первичный ключ и один или несколько индексов с целью увеличения скорости доступа к данным. Access позволяет создавать структуру таблицы в трех режимах – в режиме *конструктора*, с помощью *мастера* и путем *ввода данных*.

Разница предполагает использование этих средств пользователями с разным уровнем подготовки, разными целями и перспективами использова-

ния данных. Естественно имеется возможность просматривать, редактировать, удалять и добавлять записи, осуществлять поиск, замену, сортировку данных, изменять вид таблицы. Связи между таблицами определяются специальным средством, которое называется «Схема данных» (рис. 3).

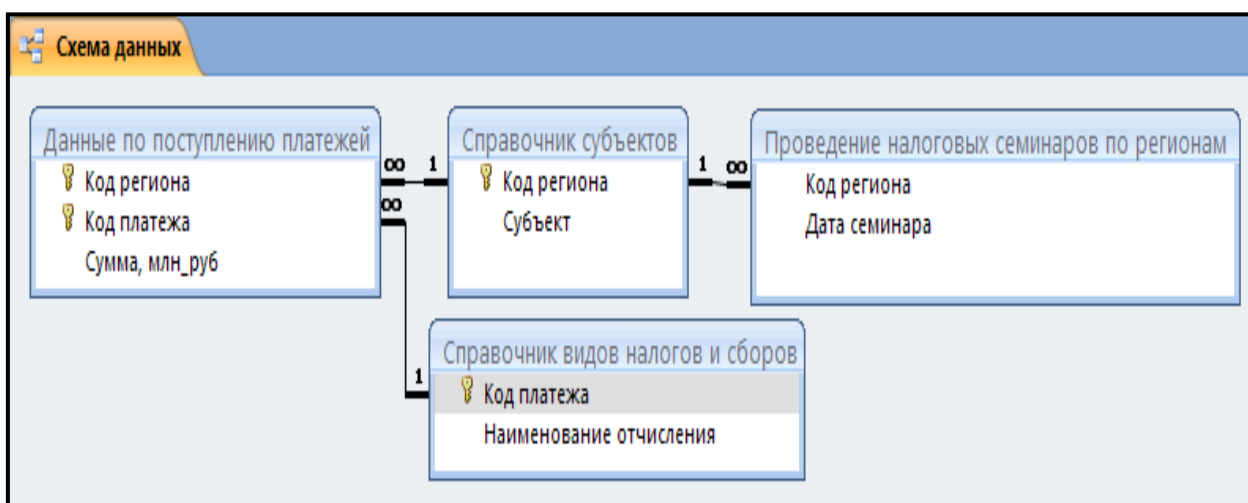


Рисунок 3. Окно Схема данных

Это удобный графический инструмент, позволяющий создавать связи между определенными полями таблиц, задавать различные типы отношений, устанавливать ограничения ссылочной целостности. При этом изменения сразу применяются в базе данных (естественно, если содержащиеся данные удовлетворяют всем условиям) Полученную диаграмму таблиц и связей можно распечатать, что, несомненно, удобно для разработчика.

Запрос – объект, содержащий текст SQL запроса, имеющий уникальное имя в определенной базе данных. Создать запрос можно с помощью мастера и в режиме конструктора. В первом случае пользователю в интерактивном режиме предлагается выбрать имя таблицы и поля для выборки. Во втором случае можно выбрать несколько таблиц или запросов, связать их графическим способом и определить поля выборки. Также можно задать дополнительные условия для каждого поля и параметры сортировки.

Есть еще один способ создания запроса, который встроен в конструктор, – это написание запроса вручную на языке SQL.

Форма – это специальный объект-контейнер для других интерфейсных компонентов, таких как поля ввода и отображения данных, кнопки и др. На форме разработчик располагает компоненты для ввода, корректировки, просмотра и группировки данных, в зависимости от специфики приложения.

Форму также можно создать двумя способами – в режиме конструктора и с помощью мастера. В первом случае разработчик располагает набором

компонентов, которые свободно размещает на форме и задает их параметры. В режиме мастера пользователь просто выбирает таблицу, поля и стиль оформления, а форма генерируется автоматически.

Отчёт – объект, предназначенный для создания документа, который впоследствии может быть распечатан либо включён в документ другого приложения. Этот документ содержит результаты выборки из базы данных в виде структурированной информации (например, в виде таблицы или списка).

MS Access обладает богатыми возможностями по оформлению и форматированию отчетов. Те же два режима создания есть и у отчета. В режиме мастера у пользователя имеется возможность задать множество параметров, практически полностью определяющих желаемый внешний вид отчета. Для создания нестандартных отчетов лучше пользоваться конструктором.

Страницы – средство публикации данных в локальной сети или Internet. Создаваемая страница проектируется подобно форме (с некоторыми отличиями в используемых компонентах), при работе с ней можно не только просматривать, но и изменять данные в базе.

После сохранения страницы как объекта в БД ее можно экспортировать в виде файла в формате HTML и использовать для доступа к данным через интернет-браузер.

Макрос – это объект, представляющий собой последовательность макрокоманд для автоматизации наиболее часто выполняемых действий при работе с базой. Макрокоманды выбираются из имеющего списка, а параметры задаются разработчиком. Выполнить макрос можно по нажатию на кнопку и программным методом в коде.

Модуль – контейнер программного кода на VBA. Для их редактирования и просмотра используется оболочка Редактора Visual Basic.

MS Access способен осуществлять взаимодействие с другими источниками и потребителями информации. Особенно прозрачно настраиваются связи с другими продуктами пакета Microsoft Office. Например, вы можете иметь документ Microsoft Word, в котором будут присутствовать поля из БД MS Access, изменение данных в базе автоматически отображается в документе. Это позволяет создавать гибкие решения, интегрирующие данные в офисных средствах. Файл, созданный средствами MS Access, может использоваться как хранилище данных.

5. Область использования

Достаточно часто, особенно если над определенной проблемой работает не отдельный специалист, а коллектив, возникает необходимость упорядочить, отсортировать накопленную информацию. В общем, создать условия, при которых можно было бы с наименьшими затратами найти нужные сведения, внести в них изменения и затем предоставить эту информацию для общего пользования.

Проанализировав возможные области применения технологии баз данных, можно выделить следующие структуры:

- применение в малом и среднем бизнесе (бухгалтерский учет, ввод заказов, ведение информации о клиентах, ведение информации о деловых контактах, кадрах и т.п.);
- при разработке программ и хранилищ данных на заказ (разработка внутриотраслевых приложений, разработка межотраслевых приложений, автоматизация некоторых функций предприятий);
- в крупных корпорациях (приложения для рабочих групп, системы обработки информации, документооборот);
- в качестве персональных СУБД (справочник по адресам, ведение инвестиционного портфеля, поваренная книга, каталоги книг, пластинок, видеофильмов и т. п.); в качестве средства хранения данных, которое используется в других приложениях.

Например, один из лидеров среди геоинформационных систем – ArcGis, создает и использует базу данных в качестве «персональной геобазы», то есть хранилища данных, где не требуется одновременное многопользовательское редактирование. Это сферы использования СУБД, хотя их конкретных реализаций может быть неизмеримо много, как и областей применения информационных технологий в целом.