Содержание

Введение

Глава1. Основы баз данных

1.1.Классификация баз данных

1.2. Функциональные возможности СУБД

1.3Модели описания баз данных

## 1.4. Основы работы настольных СУБД

1.5.Требования и стандарты, предъявляемые к базам данных

Глава 2. Работа с базой данных Microsoft Access

2.1. Основы работы настольной СУБД Microsoft Access

2.2. Работа с базой данных Microsoft Access

Заключение

Список использованной литературы

Введение

Потоки информации, циркулирующие в мире, который нас окружает, огромны. Во

времени они имеют тенденцию к увеличению. Поэтому в любой организации, как

большой, так и маленькой, возникает проблема такой организации управления

данными, которая обеспечила бы наиболее эффективную работу. Некоторые

организации используют для этого шкафы с папками, но большинство предпочитают

компьютеризированные способы – базы данных, позволяющие эффективно хранить,

структурировать и систематизировать большие объемы данных. И уже сегодня без баз

данных невозможно представить работу большинства финансовых, промышленных,

торговых и прочих организаций. Не будь баз данных, они бы просто захлебнулись в

информационной лавине.

Существует много веских причин перевода существующей информации на компьютерную основу. Сейчас стоимость хранения информации в файлах ЭВМ дешевле, чем на бумаге. Базы данных позволяют хранить, структурировать информацию и извлекать

оптимальным для пользователя образом. Данная тема актуальна в настоящее время, т.к. использование клиент/серверных технологий позволяют сберечь значительные средства, а главное и время для получения необходимой информации, а также упрощают доступ и ведение, поскольку они основываются на комплексной обработке данных и централизации их хранения. Кроме того ЭВМ позволяет хранить любые форматы данных, текст, чертежи, данные в рукописной форме, фотографии, записи голоса и т.д.

Для использования столь огромных объемов хранимой информации, помимо развития

системных устройств, средств передачи данных, памяти, необходимы средства

обеспечения диалога человек - ЭВМ, которые позволяют пользователю вводить

запросы, читать файлы, модифицировать хранимые данные, добавлять новые данные

или принимать решения на основании хранимых данных. Для обеспечения этих функций

созданы специализированные средства – системы управления базами данных (СУБД).

Целью данной работы является раскрыть понятие базы данных и системы управления базами данных, а также рассмотреть на конкретном примере работу настольной СУБД.

**1.1.Классификация баз данных**

База данных – это информационная модель предметной области, совокупность взаимосвязанных, хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений. Данные (файлы) хранятся во внешней памяти и используются в качестве входной информации для решения задач.

СУБД - это программа, с помощью которой реализуется централизованное управление данными, хранимыми в базе, доступ к ним, поддержка их в актуальном состоянии.

Системы управления базами данных можно классифицировать по способу установления связей между данными, характеру выполняемых ими функций, сфере применения, числу поддерживаемых моделей данных, характеру используемого языка общения с базой данных и другим параметрам.

Классификация СУБД:

* по выполняемым функциям СУБД подразделяются на операционные и информационные;
* по сфере применения СУБД подразделяются на универсальные и проблемно-ориентированные;
* по используемому языку общения СУБД подразделяются на замкнутые, имеющие собственные самостоятельные языки общения пользователей с базами данных, и открытые, в которых для общения с базой данных используется язык программирования, расширенный операторами языка манипулирования данными;
* по числу поддерживаемых уровней моделей данных СУБД подразделяются на одно-, двух-, трехуровневые системы;
* по способу установления связей между данными различают реляционные, иерархические и сетевые базы данных;
* по способу организации хранения данных и выполнения функций обработки базы данных подразделяются на централизованные и распределенные.

Системы централизованных баз данных с сетевым доступом предполагают две основные архитектуры – файл-сервер или клиент-сервер.

*Архитектура файл-сервер.* Предполагает выделение одной из машин сети в качестве центральной (главный сервер файлов), где хранится совместно используемая централизованная база данных. Все другие машины исполняют роль рабочих станций. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится их обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает.

*Архитектура клиент-сервер.* Эта модель взаимодействия компьютеров в сети для современных СУБД фактически стала стандартом. Каждый из подключенных к сети и составляющих эту архитектуру компьютеров играет свою роль: сервер владеет и распоряжается информационными ресурсами системы, клиент имеет возможность пользоваться ими. Помимо хранения централизованной базы данных сервер базы данных обеспечивает выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запроса SQL.

Сервер базы данных представляет собой СУБД, параллельно обрабатывающую запросы, поступившие со всех рабочих станций. Как правило, клиент и сервер территориально отделены друг от друга, и в этом случае они образуют систему распределенной обработки данных.

1.2. Функциональные возможности СУБД

Характеристиками СУБД являются:

* производительность;
* обеспечение целостности данных на уровне баз данных;
* обеспечение безопасности данных;
* возможность работы в многопользовательских средах;
* возможность импорта и экспорта данных;
* обеспечение доступа к данным с помощью языка SQL;
* возможность составления запросов;
* наличие инструментальных средств разработки прикладных программ.

Производительность СУБД оценивается:

* временем выполнения запросов;
* скоростью поиска информации;
* временем импортирования баз данных из других форматов;
* скоростью выполнения операций (таких как обновление, вставка, удаление);
* временем генерации отчета и другими показателями.
* Безопасность данных достигается:
* шифрованием прикладных программ;
* шифрованием данных;
* защитой данных паролем;
* ограничением доступа к базе данных (к таблице, к словарю и т.д.).

***Обеспечение целостности данных*** подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной. Целостность данных должна обеспечиваться независимо от того, каким образом данные заносятся в память (в интерактивном режиме, посредством импорта или с помощью специальной программы). Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и надежной безопасности.

Система управления базами данных управляет данными во внешней памяти, обеспечивает надежное хранение данных и поддержку соответствующих языков базы данных. Важной функцией СУБД является функция управления буферами оперативной памяти. Обычно СУБД работают с базами данных больших размеров, часто превышающими размеры оперативной памяти ЭВМ. В развитых СУБД поддерживается свой набор буферов оперативной памяти с собственной дисциплиной их замены.

Наибольшее распространение в настоящее время получили системы управления базами данных Microsoft Access и Oracle.

Этапами работы в СУБД являются:

* создание структуры базы данных, т.е. определение перечня полей, из которых состоит каждая запись таблицы, типов и размеров полей (числовой, текстовый, логический и т.д.), определение ключевых полей для обеспечения необходимых связей между данными и таблицами;
* ввод и редактирование данных в таблицах баз данных с помощью представляемой по умолчанию стандартной формы в виде таблицы и с помощью экранных форм, специально создаваемых пользователем;
* обработка данных, содержащихся в таблицах, на основе запросов и на основе программы;
* вывод информации из ЭВМ с использованием отчетов и без использования отчетов.

Реализуются названные этапы работы с помощью различных команд.

Централизованная база данных обеспечивает простоту управления, улучшенное использование данных на местах при выполнении дистанционных запросов, более высокую степень одновременности обработки, меньшие затраты на обработку.

Распределенная база данных предполагает хранение и выполнение функций управления данными в нескольких узлах и передачу данных между этими узлами в процессе выполнения запросов. В такой базе данных не только различные ее таблицы могут храниться на разных компьютерах, но и разные фрагменты одной таблицы. При этом для пользователя не имеет значения как организовано хранение данных, он работает с такой базой, как с централизованной.

**1.3.Модели описания баз данных**

Известны три типа моделей описания баз данных – иерархическая, сетевая и реляционная, основное различие между которыми состоит в характере описания взаимосвязей и взаимодействия между объектами и атрибутами базы данных.

*Иерархическая модель* предполагает использование для описания базы данных древовидных структур, состоящих из определенного числа уровней. «Дерево» представляет собой иерархию элементов, называемых узлами. Под элементами понимается список, совокупность, набор атрибутов, элементов, описывающих объекты.

В качестве примера простой иерархической структуры можно привести административную структуру высшего учебного заведения, элементами которой являются: «Университет – Факультет – Группа». На каждом уровне иерархии данной структуры могут быть использованы различные атрибуты. Например, атрибутами третьего уровня могут быть: специализация группы, численный состав, фамилия старосты группы и другие.

В данной модели имеется корневой узел или просто корень – «Университет», который находится на самом верхнем уровне иерархии, а потому не имеет узлов, стоящих выше его. Каждый узел модели имеет только один исходный, находящийся по отношению к нему на более высоком уровне, а на последующих уровнях классификации он может иметь один, два или большее количество узлов, либо не иметь их вообще.

Принципы иерархии:

* иерархия всегда начинается с корневой вершины (или главного узла);
* исходный узел, из которого строится дерево, называется корневым узлом или просто корнем, причем одно дерево может иметь только один корень;
* узел может содержать один или несколько атрибутов, описывающих находящийся в нем объект;
* порожденные узлы могут встраиваться в «дерево» как в горизонтальном направлении, так и в вертикальном;
* доступ к порожденным узлам возможен только через исходный узел, поэтому существует только один путь доступа к каждому узлу.

Достоинством модели является простота ее построения, легкость понимания сути принципа иерархии, наличие промышленных СУБД, поддерживающих данную модель. Недостатком является сложность операций по включению в иерархию информации о новых объектах базы данных и удалению устаревшей информации.

*Сетевая модель* описывает элементарные данные и отношения между ними в виде ориентированной сети. Это такие отношения между объектами, когда каждый порожденный элемент имеет более одного исходного и может быть связан с любым другим элементом структуры. Например, в структуре управления учебным заведением порожденный элемент «Студент» может иметь не один, а два исходных элемента: «Студент – Учебная группа», и «Студент – Комната в общежитии».

Сетевые структуры могут быть многоуровневыми и иметь разную степень сложности. Схема, в которой присутствует хотя бы одна связь «многие ко многим» и которая требует для своей реализации использования сложных методов, является сложной схемой.

База данных, описываемая сетевой моделью, состоит из областей, каждая из которых состоит из записей, а последние, в свою очередь, состоят из полей. Недостатком сетевой модели является ее сложность, возможность потери независимости данных при реорганизации базы данных. При появлении новых пользователей, новых приложений и новых видов запросов происходит рост базы данных, что может привести к нарушению логического представления данных.

*Реляционная модель* имеет в своей основе понятие «отношения», и ее данные формируются в виде таблиц. Отношение – это двумерная таблица, имеющая свое название, в которой минимальным объектом действий, сохраняющим ее структуру, является строка таблицы (кортеж), состоящая из ячеек таблицы – полей.

Каждый столбец таблицы соответствует только одной компоненте этого отношения. С логической точки зрения реляционная база данных представляется множеством двумерных таблиц различного предметного наполнения.

В зависимости от содержания отношения реляционной базы данных бывают объективными и связными. Объективные отношения хранят данные о каком-либо одном объекте, экземпляре сущности. В них один из атрибутов однозначно определяет объект и называется ключом отношения или первичным атрибутом (для удобства он записывается в первом столбце таблицы). Остальные атрибуты функционально зависят от этого ключа. В объективном отношении не может быть дублирующих объектов и в этом – основное ограничения реляционной базы данных. Связное отношение хранит ключи нескольких объектных отношений, по которым между ними устанавливаются связи.

Если набор атрибутов базы данных заранее не фиксирован, то возможны различные варианты их группировки, однако, независимо от выбранного способа, должны соблюдаться единые требования. В частности, если база данных содержит множество отношений, то они должны иметь минимальную избыточность представления информации; атрибуты, включаемые в базу данных, должны обеспечивать выполнение массовых расчетов; при добавлении в базу данных новых атрибутов перестройка наборов отношений должна быть минимальной.

К числу достоинств реляционной модели относятся: простота построения, доступность понимания, возможность эксплуатации базы данных без знания методов и способов ее построения, независимость данных, гибкость структуры и другие. Недостатками модели являются: низкая производительность по сравнению с иерархической и сетевой моделями, сложность программного обеспечения, избыточность.

## **1.4. Настольные СУБД**

Многие авторы классифицируют СУБД на две большие категории: «настольные» и «серверные».

Настольные СУБД отличаются тем, что используют в модель вычислений с сетью и файловым сервером (архитектура «файл-сервер»). Увеличение сложности задач, появление персональных компьютеров и локальных вычислительных сетей явилось предпосылками появления новой архитектуры «файл-сервер». Эта архитектура баз данных с сетевым доступом предполагает назначение одного из компьютеров сети в качестве выделенного сервера, на котором будут храниться файлы базы данных. В соответствие с запросами пользователей файлы с файл-сервера передаются на рабочие станции пользователей, где и осуществляется основная часть обработки данных. Центральный сервер выполняет в основном только роль хранилища файлов, не участвуя в обработке самих данных.

Работа построена следующим образом:

База данных в виде набора файлов находится на жестком диске специально выделенного компьютера (файлового сервера). Существует локальная сеть, состоящая из клиентских компьютеров, на каждом из которых установлены СУБД и приложение для работы с БД. На каждом из клиентских компьютеров пользователи имеют возможность запустить приложение. Используя предоставляемый приложением пользовательский интерфейс, он инициирует обращение к БД на выборку/обновление информации.

Все обращения к БД идут через СУБД, которая инкапсулирует внутри себя все сведения о физической структуре БД, расположенной на файловом сервере. СУБД инициирует обращения к данным, находящимся на файловом сервере, в результате которых часть файлов БД копируется на клиентский компьютер и обрабатывается, что обеспечивает выполнение запросов пользователя (осуществляются необходимые операции над данными). При необходимости (в случае изменения данных) данные отправляются назад на файловый сервер с целью обновления БД. Результат СУБД возвращает в приложение. Приложение, используя пользовательский интерфейс, отображает результат выполнения запросов. В рамках архитектуры «файл-сервер» были выполнены первые версии популярных т.н. настольных СУБД, таких как dBase и Microsoft Access.

Указываются следующие основные недостатки данной архитектуры: при одновременном обращении множества пользователей к одним и тем же данным производительность работы резко падает, т.к. необходимо дождаться пока пользователь, работающий с данными, завершит свою работу. В противном случае возможно затирание исправлений, сделанных одними пользователями, изменениями других пользователей. [2, 78]

На сегодняшний день известно более двух десятков форматов данных настольных СУБД, однако наиболее популярными, исходя из числа проданных копий, следует признать dBase, Paradox, FoxPro и Access. Из появившихся недавно СУБД следует также отметить Microsoft Data Engine - по существу серверную СУБД, представляющую собой <облегченную> версию Microsoft SQL Server, но предназначенную, тем не менее, для использования главным образом в настольных системах и небольших рабочих группах.

|  |  |  |
| --- | --- | --- |
| СУБД | Производитель | URL |
| Visual dBase | dBase, Inc | http://www.dbase2000.com/ |
| Paradox | Corel | http://www.corel.com/ |
| Microsoft Access 2000 | Microsoft | http://www.microsoft.com/ |
| Microsoft FoxPro | Microsoft | http://www.microsoft.com/ |
| Microsoft Visual FoxPro | Microsoft | http://www.microsoft.com/ |
| Microsoft Visual FoxPro | Microsoft | http://www.microsoft.com/ |
| Microsoft Data Engine | Microsoft | http://www.microsoft.com/ |

**1.5.Требования и стандарты, предъявляемые к базам данных**

К современным базам данных, а, следовательно, и к СУБД, на которых они строятся, предъявляются следующие основные требования:

* Высокое быстродействие (малое время отклика на запрос).
* Простота обновления данных.
* Независимость данных.
* Совместное использование данных многими пользователями.
* Безопасность данных - защита данных от преднамеренного или непреднамеренного нарушения секретности, искажения или разрушения.
* Стандартизация построения и эксплуатации БД (фактически СУБД).
* Адекватность отображения данных соответствующей предметной области.
* Дружелюбный интерфейс пользователя.

Важнейшими являются первые два противоречивых требования: повышение быстродействия требует упрощения структуры БД, что, в свою очередь, затрудняет процедуру обновления данных, увеличивает их избыточность. Независимость данных - возможность изменения логической и физической структуры БД без изменения представлений пользователей. Целостность данных - устойчивость хранимых данных к разрушению и уничтожению, связанных с неисправностями технических средств, системными ошибками и ошибочными действиями пользователей.

 Стандартизация обеспечивает преемственность поколений СУБД, упрощает взаимодействие БД одного поколения СУБД с одинаковыми и различными моделями данных. Стандартизация (ANSI/SPARC) осуществлена в значительной степени в части интерфейса пользователя СУБД и языка SQL. Это позволило успешно решить задачу взаимодействия различных реляционных СУБД как с помощью языка SQL, так и с применением приложения Open DataBase Connection (ODBC). При этом может быть осуществлен как локальный, так и удаленный доступ к данным (технология клиент/сервер или сетевой вариант).

**Глава 2. Работа с базой данных Microsoft Access**

**2.1. Основы работы настольной СУБД Microsoft Access**

Первая версия СУБД Access появилась в начале 90-х годов. Это была первая настольная реляционная СУБД для 16-разрядной версии Windows. Популярность Access значительно возросла после включения этой СУБД в состав Microsoft Office.

В отличие от Visual FoxPro, фактически превратившегося в средство разработки приложений, Access ориентирован в первую очередь на пользователей Microsoft Office, в том числе и не знакомых с программированием. Это, в частности, проявилось в том, что вся информация, относящаяся к конкретной базе данных, а именно таблицы, индексы (естественно, поддерживаемые), правила ссылочной целостности, бизнес-правила, список пользователей, а также формы и отчеты хранятся в одном файле, что в целом удобно для начинающих пользователей.

Популярность СУБД MS Access обусловлена следующими причинами:

* Access является одной из самых легкодоступных и понятных систем как для профессионалов, так и для начинающих пользователей, позволяющая быстро освоить основные принципы работы с базами данных;
* Система имеет полностью русифицированную версию;
* Полная интегрированность с пакетами MS Office: Word, Excel, Power Point, Mail;
* Возможность использования OLE технологии, что позволяет установить связь с объектами другого приложения или внедрить какие-либо объекты в базу данных Access;
* Широко и наглядно представлена справочная система;
* Существует набор «мастеров» по разработке объектов, облегчающий создание таблиц, форм и отчетов

Поддержка COM в Access выражается в возможности использовать элементы управления ActiveX в формах и Web-страницах, созданных с помощью Access. В отличие от Visual FoxPro создание COM-серверов с помощью Access не предполагается.

Иными словами, Microsoft Access может быть использован, с одной стороны, в качестве настольной СУБД и составной части офисного пакета, а с другой стороны, в качестве клиента Microsoft SQL Server, позволяющего осуществлять его администрирование, манипуляцию его данными и создание приложений для этого сервера.

Помимо манипуляции данными Microsoft SQL Server, Access 2000 позволяет также в качестве хранилища данных использовать Microsoft Data Engine (MSDE), представляющий собой по существу настольный сервер баз данных, совместимый с Microsoft SQL Server.

**2.2. Работа с базой данных Microsoft Access**

Создание пустой базы данных.

Для создания новой базы данных выполните команду Файл/Создать. На экране откроется окно диалога “Создание”, содержащее две вкладки:

“Общие” — позволяет создать новую пустую базу данных;

“Базы данных” — позволяет выбрать образец базы данных, содержащий большинство требуемых объектов, и создать базу с помощью мастера.

Для создания новой пустой базы данных перейдите на вкладку “Общие” и нажмите кнопку ОК в нижней части окна диалога. На экране откроется окно диалога “Файл новой базы данных” . В данном окне диалога из раскрывающегося списка Папка выберите папку, в которой хотите сохранить создаваемую базу данных, а в поле ввода Имя файла введите имя базы данных. После этого Вы можете создать с помощью мастера базу данных определенного типа со всеми необходимыми таблицами, формами и отчетами. Так как MS Access содержит большой выбор подготовленных для вас баз данных, второй способ во многих случаях может оказаться предпочтительным. В обоих случаях у Вас останется возможность в любое время изменить и расширить созданную Вами базу данных.

При выборе имени базы данных не ограничивайте себя восемью символами, а дайте волю фантазии. Расширение для имени файла указывать не нужно, по умолчанию в поле ввода Тип файла установлен тип “База данных”. После ввода имени создаваемой базы данных нажмите кнопку Создать данного окна диалога. На экране откроется окно базы данных. Оно состоит из шести вкладок, которые пока пусты. В данном окне Вам предстоит создать все объекты, входящие в базу данных. Их перечень соответствует ярлыкам вкладок в верхней части окна базы данных. В этом окне Вы можете создавать таблицы, хранящие информацию, отчеты, формы, запросы. Все они будут располагаться во вкладках, которые открываются при выборе соответствующего ярлыка.

Создание базы данных с помощью мастера.

Для создания новой базы данных с помощью мастера выполните следующие действия:

Выполните команду Файл [Создать]

В открывшемся окне диалога “Создание” выберите ярлык “Базы данных”. На экране появится список баз данных, предлагаемых мастером. Данный список очень велик и может достигать нескольких десятков различных вариантов, которые могут сразу использоваться или послужат основой для построения других баз данных. Например, “Заказы на работы”, “Счета”, “Контакты”, “Мероприятия”, … и т.п.

Выберите из списка подходящий Вам образец базы данных и запустите на выполнение мастер создания базы данных, нажав кнопку ОК.

В открывшемся окне диалога “Файл новой базы данных” из раскрывающегося списка Папка выберите папку, в которой хотите сохранить создаваемую базу данных, а в поле Имя файла введите ее имя. Затем нажмите кнопку Создать.

В следующем окне диалога мастер сообщает, какую информацию будет содержать создаваемая им база данных. В нижней части этого окна диалога находятся следующие кнопки:

•Отмена — прекращает работу мастера;

•Назад — позволяет вернуться к предыдущему шагу в работе мастера;

•Далее — позволяет перейти к следующему шагу в работе мастера;

•Готово — запускает мастер создания базы данных с выбранными параметрами, причем перед нажатием этой кнопки высвечивается информация, которая будет храниться в базе данных.

Для продолжения работы нажмите кнопку Далее.

Открывшееся окно диалога содержит два списка. Первый из них — список таблиц базы данных, а второй — список полей выбранной таблицы. В этом списке отмечены поля, которые будут включены в таблицу. Обычно отмечены почти все поля таблиц, за исключением полей, которые используются достаточно редко. Устанавливая или снимая флажки для полей, Вы можете выбрать поля таблицы. После того как выбрали поля таблиц, нажмите кнопку Далее.

В следующем окне диалога выберите из предлагаемых образцов вид оформления экрана и нажмите кнопку Далее (при этом на экране Вам предлагается возможность предварительного просмотра видов оформления экранов, которые Вы можете перебирать в правом окне окна диалога).

На следующем шаге работы мастера можно определить вид создаваемых для базы данных отчетов . После выбора подходящего вам вида нажмите кнопку Далее (здесь также Вам предлагается предварительно ознакомиться с возможными вариантами, которые также можно перебирать).

Открывшееся затем окно диалога мастера создания базы данных позволяет задать ее заголовок и рисунок (например, торговый знак фирмы), который будет появляться во всех отчетах. Если Вы решили использовать рисунок, установите флажок Да. В этом случае становится доступна кнопка Рисунок, нажатие на которую открывает окно диалога “Выбор рисунка” для выбора заранее созданного Вами файла с рисунком. Нажмите кнопку Далее для выполнения дальнейших установок.

Нажав кнопку Готово в последнем окне диалога, Вы запускаете мастер на построение базы данных с установленными параметрами. Используя кнопку Назад, Вы можете вернуться на любой из предыдущих шагов и изменить параметры базы данных. Вы можете нажать кнопку Готово в любом окне диалога мастера, отказавшись от дальнейшей установки дополнительных параметров. В этом случае мастер в своей работе использует установки, принимаемые по умолчанию.

После нажатия кнопки Готово мастер переходит к созданию базы данных, состоящей из таблиц с заданными Вами полями, простейших форм ввода и просмотра информации и простейших отчетов. После завершения процесса создания базы данных Вы сразу же можете воспользоваться готовой базой данных: вводить в таблицы данные, просматривать их и распечатывать.

Если варианты предложенных баз данных Вас не устраивают, то Вы можете создать пустую базу данных и добавить в нее таблицы, запросы, формы и отчеты[[1]](#footnote-1).

Создание таблиц.

Итак, Вы приступаете к созданию таблиц базы данных, в которые впоследствии будет вводиться информация. В дальнейшем данные в таблице могут дополняться новыми данными, редактироваться или исключаться из таблицы. Вы можете просматривать данные в таблицах или упорядочивать их по некоторым признакам. Информация, содержащаяся в таблицах, может быть использована для составления отчетов. Кроме того, Вы можете дать графическую интерпретацию информации, содержащейся в базе данных. С решением этих задач Вы познакомитесь в последующих главах.

Создание таблицы в MS Access осуществляется в окне базы данных. Рассмотрим последовательность Ваших действий при создании таблицы в новой базе данных:

Откройте окно созданной Вами базы данных и перейдите на вкладку “Таблицы”.

Нажмите кнопку Создать в окне базы данных.

Откроется окно диалога “Новая таблица”, в правой части которого находится список вариантов дальнейшей работы:

Режим таблицы — позволяет создать новую таблицу в режиме таблицы;

Конструктор — позволяет создать новую таблицу в конструкторе таблиц;

Мастер таблиц — позволяет создать новую таблицу с помощью мастера;

Импорт таблиц — позволяет осуществить импорт таблиц из внешнего файла в текущую базу данных;

Связь с таблицами — позволяет осуществить создание таблиц, связанных с таблицами из внешних файлов.

Выберите из этой таблицы подходящий Вам вариант создания таблицы и нажмите кнопку ОК.

Создайте структуру таблицы с помощью выбранного Вами средства. Создать таблицу можно с помощью мастера и с помощью конструктора.

Для того чтобы связать таблицу с содержащейся в ней информацией, каждой таблице присваивается имя. Задайте имя таблицы в окне диалога “Сохранение” и нажмите кнопку ОК.

При присвоении имени таблице, как и имени базы данных, Вы можете не ограничиваться восемью символами. Имя таблицы, как и имена других объектов базы данных, хранится в самой базе данных[[2]](#footnote-2).

Наименование поля

Наименование поля вводится в поле ввода столбца имя поля. При задании наименований полей Вы должны следовать следующим правилам:

Наименование поля может содержать до 64 символов, но не следует злоупотреблять этой возможностью, задавая слишком длинные имена;

Наименование поля может содержать буквы, цифры, пробелы и специальные символы, за исключением точки (.), восклицательного знака (!), прямых скобок ([]) и некоторых управляющих символов (с кодами ASCII 0-31);

Наименование поля не может начинаться с пробела;

Два поля в одной таблице не могут иметь одинаковых наименований;

В качестве наименования поля не рекомендуется использовать аббревиатуры или краткие названия.

Несоблюдение этих правил отслеживается средствами СУБД MS Access, но в некоторых случаях это может привести к трудно определяемым ошибкам, поэтому рекомендуется самостоятельно контролировать следование вышеперечисленным правилам в практической работе.

Желательно стараться использовать имена, отличающиеся краткостью, для облегчения их идентификации при просмотре таблиц.

Типы данных

Наименование поля используется для ссылки на данные таблицы. Для определения типа хранимых данных используется тип данных. Тип данных поля вводится в поле ввода столбца Тип данных. В MS Access допустимыми являются данные следующих типов:

* Текстовый;
* Числовой;
* Денежный;
* Счетчик;
* Даты/времени;
* Логический;
* Поле MEMO:
* Поле объекта OLE;
* Мастер подстановок.

Текстовые поля могут содержать буквы, цифры и специальные символы. Максимальная ширина поля составляет 255 символов.

Для изменения ширины поля нужно в строке Размер поля раздела “Свойства поля” задать число, определяющее ширину поля (от 1 до 255).

Каждый из типов данных наделен собственными свойствами, которые отображаются в разделе “Свойства поля” окна конструктора.

Создание структуры таблицы

После того как мы рассмотрели типы данных в Access и отдельные свойства полей таблицы, можно приступить к созданию структуры таблицы. Создание структуры таблицы рассмотрим на примере создания таблицы Заказы базы данных Борей, поставляемой вместе с Access. Данная таблица уже имеется в базе данных Борей, но рассмотрение процесса создания этой таблицы будет весьма полезным. Для того, чтобы не нарушить структуру базы данных Борей, предварительно, создайте учебную базу данных и откройте ее окно.

В окне конструктора таблицы в столбце Имя поля введите КодЗаказа.

Нажмите клавишу Tab или Enter, чтобы перейти в столбец Тип данных. При этом обратите внимание, что в нижней части окна диалога появляется информация в разделе “Свойства поля”.

В столбце Тип данных появилось значение Текстовый. Нажмите на кнопку раскрытия списка в правой части прямоугольника, и Вы увидите список, содержащий все типы данных. Из этого списка с помощью мыши или клавишами вверх и вниз выберите значение Счетчик и нажмите на клавишу Tab для перехода в столбец Описание. Столбец Описание представляет собой пояснение, которое Вы даете своим полям. Когда Вы в будущем будете работать с данной таблицей, это описание будет появляться в нижней части экрана MS Access всякий раз, когда Вы окажетесь в поле КодЗаказа, и напомнит Вам назначение данного поля.

Введите поясняющий текст в столбец Описание и нажмите клавишу Tab или Enter, чтобы перейти к вводу информации о следующем поле.Аналогичным образом введите описание всех полей таблицы.Завершив ввод структуры таблицы, сохраните ее, выполнив команду Файл/Сохранить.

Создание таблицы в режиме таблицы

Мы рассмотрели два способа создания таблиц, теперь переходим к третьему. Каждый из способов имеет свои преимущества и недостатки. Но мы уверены, что этот способ создания таблицы поразит Вас своей простотой, наглядностью и очень понравится. Не зря он расположен первым в списке методов создания таблиц окна диалога “Новая таблица”. Вы, вероятно, им чаще всего и будете пользоваться.

Ниже приведена последовательность действий, которую Вам предстоит выполнить:

Перейдите на вкладку “Таблицы” окна базы данных и нажмите кнопку Создать.

В окне диалога “Новая таблица” выберите из списка вариантов значение Режим таблицы и нажмите кнопку ОК. В результате выполнения этих действий откроется окно диалога “Таблица”, содержащее созданную по умолчанию таблицу. Эта таблица содержит 20 столбцов и 30 строк, и этого вполне достаточно для начала. После сохранения этой таблицы Вы, конечно, можете добавить столько строк и столбцов, сколько Вам понадобится.

Наименования полей таблицы определены по умолчанию, но вряд ли они удовлетворяют Вашим требованиям. MS Access позволяет очень просто присвоить полям новые имена. Для этого нажмите дважды кнопкой мыши на область выбора первого поля (заголовок которого содержит Поле 1). Имя поля выделяется и появляется мигающий курсор. Введите имя первого поля и нажмите клавишу Tab. Аналогично введите остальные имена полей вашей таблицы в следующих столбцах.

Теперь заполните несколько строк Вашей таблицы, вводя информацию в том виде, в каком она будет вводиться и в будущем. Старайтесь записывать все в одном стиле (например, если первую дату Вы записали 10/14/09, то не пишите следующую в виде Ноябрь 3, 2009). Если MS Access установит неправильный тип данных, Вы сможете его изменить, но лучше вводить все правильно сразу.

Сохраните таблицу, выполнив команду Файл/Сохранить макет или нажав кнопку Сохранить на панели инструментов. В открывшемся окне диалога “Сохранение” присвойте таблице имя и нажмите кнопку ОК.

На запрос о необходимости создания для таблицы первичного ключа нажмите кнопку Да, и MS Access создаст таблицу, удалив лишние строки и столбцы.

Теперь убедитесь, что Access выбрал для каждого поля правильные типы данных. Для этого перейдите в окно конструктора таблицы, выполнив команду Вид/Конструктор таблиц. Если Вас что-то не устраивает в структуре таблицы, внесите необходимые изменения[[3]](#footnote-3).

Выборка данных

В предыдущей главе мы рассмотрели использование фильтров, а в этой главе рассмотрим более мощное средство выборки данных — запросы.

На практике часто требуется выбрать из исходной таблицы часть записей, удовлетворяющих определенным критериям, и упорядочить выборку. Критерии могут определяться сочетанием ряда условий. Например, Вам необходимо выбрать записи о поставщиках определенного товара из Минска и упорядочить их в алфавитном порядке по наименованию фирм. Для решения таких задач предназначены мастера запросов и конструктор запросов, при помощи которых Вы сможете:

Формировать сложные критерии для выбора записей из одной или нескольких таблиц;

Указать поля, отображаемые для выбранных записей;

Выполнять вычисления с использованием выбранных данных.

Что такое “Запрос по образцу”

Ранее нами были рассмотрены общие положения, относящиеся к базам данных, где мы отмечали, что одним из основных назначений баз данных является быстрый поиск информации и получение ответов на разнообразные вопросы. Вопросы, формулируемые по отношению к базе данных, называются запросами. В MS Access для формирования запросов используется конструктор запросов и инструкция SELECT языка MS Access.

Что же такое “Запрос по образцу”? Запрос по образцу — это интерактивное средство для выбора данных из одной или нескольких таблиц. При формировании запроса Вам необходимо указать критерии выборки записей в исходной таблице. При этом вместо того, чтобы печатать предложения на специальном языке, Вы должны просто заполнить бланк запроса, который располагается в окне конструктора запросов. Метод формирования запроса путем заполнения бланка прост для изучения и понимания. Он способствует эффективному использованию возможностей MS Access пользователями, имеющими даже минимальный навык работы с приложением или не имеющими его вовсе[[4]](#footnote-4).

Для создания простейших запросов Вы можете использовать мастер запросов, который у Вас последовательно запрашивает наименования таблиц, используемых в запросе, перечень полей таблиц и некоторые дополнительные параметры. Для создания более сложных запросов Вам может понадобиться конструктор запросов, который является простым и наглядным средством создания достаточно сложных запросов

Создание простого запроса

Предположим, Вы решили позвонить всем Вашим клиентам и сообщить им о презентации новых товаров. Номера телефонов находятся в одном из столбцов таблицы, содержащей всю информацию о Ваших клиентах — адреса и т.д. В данном случае Вам не требуется вся эта информация. Достаточно названия фирмы, имени представителя и номера телефона, то есть Вам нужны только три столбца. Рассмотрим создание такого запроса.

Вся необходимая информация находится в таблице Клиенты базы данных Борей. Поэтому для создания запроса выполните следующие действия:

В окне базы данных перейдите на вкладку “Запросы” и нажмите кнопку Создать.

Откроется окно диалога “Добавление таблицы”, в котором выберите таблицу Клиенты и нажмите кнопку Добавить. Закройте окно диалога.

На экране открывается окно конструктора запросов, схема данных которого содержит всего одну таблицу Клиенты, а бланк запроса пуст.

Добавление полей в бланк запроса

Для выбора полей, которые должны присутствовать в результирующей таблице, Вам необходимо отобразить их в бланке запроса.

В MS Access существует два варианта выбора полей результирующей таблицы. Вы можете воспользоваться наиболее приемлемым с Вашей точки зрения:

Для добавления в таблицу отдельных полей Вы можете выбрать поле таблицы на схеме данных и дважды нажать кнопку мыши. Выбранное поле будет вставлено в следующий доступный столбец в строке Поле бланка запроса. В строке Имя таблицы сразу же появится имя таблицы, а позиция Вывод на экран будет помечена.

В MS Access широко используется механизм перенести-и-оставить (drag-and-drop). Для использования этого механизма при выборе полей перейдите в таблицу в схеме данных, из которой Вам надо выбрать поля. Выделите поля, которые Вы собираетесь отобразить в запросе, нажмите кнопку мыши и, не отпуская ее, перенесите выбранные поля в бланк запроса

Изменение порядка полей

Порядок полей в бланке запроса определяет порядок появления их в результирующей таблице. Для того чтобы изменить расположение поля в этом списке, выполните следующие действия:

Установите указатель мыши на область выбора столбца, который располагается прямо над названием поля. Когда указатель изменит вид на стрелку, нажмите кнопку, чтобы выделить столбец.

Нажмите и удерживайте кнопку мыши в этом положении. На конце указателя появится прямоугольник.

Перемещайте столбец в требуемом направлении. Толстая вертикальная линия покажет его текущее положение.

Отпустите кнопку, когда толстая вертикальная линия окажется в требуемом месте. Поле будет перемещено в новое место.

Перемещение столбца бланка запроса. Иногда не сразу удается выделить столбец для его перемещения. Убедитесь в том, что Вы нажимаете область выбора столбца (маленький прямоугольник, который находится прямо над названием поля). Это единственное место, за которое Вы можете захватить столбец для переноса. Для переноса поля в бланк запроса дважды нажмите мышью на названии поля

Сохранение запроса.

Созданный запрос можно использовать в дальнейшем. Для этого Вы должны присвоить ему имя и сохранить его. Сохранение запроса осуществляется командой Файл/Сохранить как экспорт, который откроет окно диалога ввода имени запроса. MS ACCESS по умолчанию предложит имя запроса и поля ввода, но лучше подобрать, что-нибудь более значимое.

Для создания самого простого отчета в MS Access достаточно выполнить всего лишь несколько действий:

Откройте окно базы данных.

В окне базы данных перейдите на вкладку “Отчеты”.

Установите указатель на таблицу, для которой хотите создать отчет.

Выполните команду Вставка/Автоотчет или нажмите кнопку Новый объект на панели инструментов и из раскрывающегося списка выберите Автоотчет.

На экране в считанные секунды появится готовый к использованию отчет. В этот отчет включены все поля таблицы. Их названия расположены вертикально в том же порядке, в каком они находятся в таблице. Справа от названия каждого поля отображается его значение в таблице.

Заключение

В данной курсовой работе было рассмотрено понятие базы данных и системы управлениями базами данных, с помощью которой реализуется централизованное управление данными, хранимыми в базе, доступ к ним, поддержка их в актуальном состоянии. Итак, база данных - это совокупность взаимосвязанных данных, совместно хранимых в одном или нескольких компьютерных файлах. А также было рассмотрена работа с одной из популярных СУБД Microsoft Access.

СУБД Microsoft Access предоставляет необходимые средства для работы с базами данных неискушенному пользователю, позволяя ему легко и просто создавать базы данных, вводить в них информацию, обрабатывать запросы и формировать отчеты. К сожалению, встроенная система помощи недостаточно понятно объясняет начинающему пользователю порядок работы, поэтому возникает необходимость в пособии.

1. Марченко А. П. Microsoft Access: Краткий курс. – СПб.: Питер, 2005. – стр. 56 [↑](#footnote-ref-1)
2. Лазарев И.П.. “Microsoft Access для чайников”.. СПб – Питер, 2004. – стр.139. [↑](#footnote-ref-2)
3. Лазарев И.П.. “Microsoft Access для чайников”.. СПб – Питер, 2004. – стр. 196. [↑](#footnote-ref-3)
4. Марченко А. П. Microsoft Access : Краткий курс. – СПб.: Питер, 2005. – 239. [↑](#footnote-ref-4)