

Министерство высшего и среднего специального образования
Республики Узбекистан
Ташкентский институт текстильной и лёгкой промышленности

Кафедра: «Информатики и информационных технологий»

Курсовая работа

на тему: «Создание базы данных в Microsoft Access»

Выполнила: Маматкаримова Н.

Гр. 3р-15

Ташкент 2016 г.

СОДЕРЖАНИЕ

Введение.....	3
1 Основные понятия баз данных.....	5
1.1 Базы данных и системы управления базами данных.....	5
1.2 Состав СУБД.....	6
1.3 Свойства полей базы данных.....	6
1.4 Типы данных.....	8
1.5 Безопасность баз данных.....	9
2 Проектирование баз данных.....	10
2.1 Режимы работы с базами данных.....	10
3 Объекты базы данных.....	10
3.1 Таблицы.....	11
3.2 Запросы.....	11
3.3 Формы.....	11
3.4 Отчёты.....	12
3.5 Страницы.....	12
3.6 Макросы и Модули.....	12
4 Понятие реляционной модели и нормализация.....	14
4.1 Реляционная модель.....	14
4.2 Нормализация.....	14
4.3 Ключи.....	14
4.4 Связи.....	14
5 Построение базы данных о студентах.....	16
Заключение.....	24
Список использованной литературы.....	25
Приложение.....	26

ВВЕДЕНИЕ

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД). Существует множество различных СУБД: IMS, SETOP, ORACLE, CLIPPER, ACCESS. В данной работе мы рассмотрим только MICROSOFT ACCESS.

Увеличение объема и структурной сложности хранимых данных, расширение круга пользователей информационных систем привели к широкому распространению наиболее удобных и сравнительно простых для понимания реляционных (табличных) СУБД. Для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга и от места хранения баз данных, созданы сетевые мультиспользовательские версии БД основанных на реляционной структуре. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных, а также санкционирования доступа.

Цель курсовой работы: ознакомление с основными средствами Microsoft Access для создания баз данных; создание собственную базу данных.

Перед тем как приступить к изучению СУБД ACCESS рассмотрим несколько основных понятий.

Данные - систематизированная и структурированная каким-либо образом информация. Существуют, по крайней мере, три модели данных: иерархическая, сетевая, реляционная.

Модели данных - представление данных и их взаимосвязей, описывающих понятия проблемной среды. Модели данных используются как

для концептуального, так и для логического и физического представления данных.

В данной работе мы будем рассматривать только логическое представление данных, т.е. с точки зрения прикладного программиста и пользователя. С их точки зрения данные делятся на записи, те в свою очередь делятся на поля, и между полями, а так же и между записями могут быть установлены различные связи.

Поле - основной и минимальный элемент данных.

Запись - несколько полей.

Связи - логические взаимосвязи между записями или полями.

1. Иерархически структурированные данные состоят из нескольких записей, одна из которых определена как корневая. Между записями в иерархии могут быть определены связи: “один ко многим” или “один к одному”.

2. Сетевая модель подобна иерархической, но является более общей моделью, в том смысле, что между записями присутствуют связи “много ко многим” и “много к одному”, а так же здесь нет корневого узла, т.е. любая запись может быть корневой.

3. Реляционная модель данных является одним из самых простых видов представления информации и самой легкой концепцией для понимания. Сетевая модель и иерархическая могут быть приведены к реляционной.

В зависимости от рассмотренных выше моделей данных базы данных могут быть иерархическими, сетевыми и реляционными. Дальше мы будем рассматривать только реляционные базы данных.

1 ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

1.1 Базы данных и системы управления базами данных

База данных – это организованная структура, предназначенная для хранения информации. В современных базах данных хранятся не только данные, но и информация.

Это утверждение легко пояснить, если, например, рассмотреть базу данных крупного банка. В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состоянии расчетных счетов, финансовых операциях и т.д. Доступ к этой базе данных имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ ко всей базе полностью и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит методы и средства, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием базы данных тесно связано понятие системы управления базой данных. Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнение ее содержимым, редактирование содержимого и визуализации информации. Под визуализацией информации базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передачи по каналам связи.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство

СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта мы выберем СУБД Microsoft Access, входящую в пакет Microsoft Office.

1.2 Состав СУБД.

Язык описания данных (ЯОД) – Средства описания данных в БД и связей между ними. Средствами этого языка описывается структура БД, форматы записей, пароли, защищающие данные.

Язык манипулирования данными (ЯМД) – язык для выполнения операций над данными, позволяющий менять их строение.

Для различных СУБД реализация этих уровней языков может быть различной. В одних случаях ЯОД и ЯМД требует составления пользователем программы полностью “вручную”, в других (что отражает современную тенденцию) в СУБД присутствует средства визуальной (зримой, наглядной) разработки программ. Для этого в современных СУБД имеются редакторы экранных форм, отчетов. “Кирпичиками” (инструментами) таких редакторов являются поля различных видов (поля ввода, поля вывода, вычисляемые поля), процедуры обработки различных типов (формы ввода, таблицы, отчеты, запросы). На основании созданных пользователем объектов программы – генераторы формируют программный код на языке конкретной машины или на промежуточном языке.

1.3 Свойства полей базы данных

Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Microsoft Access.

Имя поля – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).

Тип поля – определяет тип данных, которые могут содержаться в данном поле.

Размер поля – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.

Формат поля – определяет способ форматирования данных в ячейках, принадлежащих полю.

Маска ввода – определяет форму, в которой вводятся данные, а поле (средство автоматизации ввода данных).

Подпись – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля).

Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).

Условие на значение – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных.

Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы.

Пустые строки – свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например текстовым).

Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значение в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видео клипов и других объектов OLE, большинство вышеуказанных свойств не имеют смысла.

1.4 Типы данных

Таблицы баз данных, как правило, допускают работу с большим количеством разных типов данных. Так, например, базы данных Microsoft Access работают со следующими типами данных.

Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

Числовой – тип данных для хранения действительных чисел.

Поле Мемо – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он храниться в другом месте базы данных, а в поле храниться указатель на него, но для пользователя такое разделение заметно не всегда.

Дата/время – тип данных для хранения календарных дат и текущего времени.

Денежный - тип данных для хранения денежных сумм. Теоретически, для их записи можно было бы пользоваться и полями числового типа, но для денежных сумм есть некоторые особенности (например, связанные с правилами округления), которые делают более удобным использование специального типа данных, а не настройку числового типа.

Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.

Логический - тип для хранения логических данных (могут принимать только два значения, например Да или Нет).

Гиперссылка – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать их из раскрывающегося списка.

1.5 Безопасность баз данных

Базы данных – это тоже файлы, но работа с ними отличается от работы с файлами других типов, создаваемых прочими приложениями. Выше мы видели, что всю работу по обслуживанию файловой структуры берет на себя операционная система. Для базы данных предъявляются особые требования с точки зрения безопасности, поэтому в них реализован другой подход к сохранению данных.

Базы данных – это особые структуры. Информация, которая в них содержится, очень часто имеет общественную ценность. Нередко с одной и той же базой работают тысячи людей по всей стране. От информации, которая содержится в некоторых базах, может зависеть благополучие множества людей. Поэтому целостность содержимого базы не может и не должна зависеть ни от конкретных действий некоего пользователя, забывшего сохранить файлы перед выключением компьютера, ни от перебоев в электросети.

Проблема безопасности баз данных решается тем, что в СУБД для сохранения информации используется двойной подход. В части операций, как обычно, участвует операционная система компьютера, но некоторые операции сохранения происходят в обход операционной системы.

2 ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

2.1 Режимы работы с базами данных

Обычно с базами данных работают две категории пользователей.

Первая категория – проектировщики. Их задача состоит в разработке структуры таблиц базы данных и согласование ее с заказчиком. Кроме таблиц проектировщики разрабатывают и другие объекты базы данных, предназначенные, с одной стороны, для автоматизации работы с базой, а с другой стороны – для ограничения функциональных возможностей работы с базой (если это необходимо из соображений безопасности). Проектировщики не наполняют базу конкретными данными, (заказчик может считать их конфиденциальными и не предоставлять посторонним лицам). Исключение составляет экспериментальное наполнение модельными данными на этапе отладки объектов базы.

Вторая категория - исполнителей, работающих с базами данных, - пользователи. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы – только к данным, да и то не ко всем, а к тем, работа с которыми предусмотрена на конкретном рабочем месте.

Соответственно СУБД имеет два режима работы: **проектировочный** и **пользовательский**. Первый режим предназначен для создания или изменения структуры базы, а так же создание ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

3 ОБЪЕКТЫ БАЗЫ ДАННЫХ.

3.1 Таблицы

Таблицы – это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства).

3.2 Запросы

Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразования данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнения таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

3.3 Формы

Если **запросы** – это специальные средства для отбора и анализа данных, то **формы** – это средства для ввода данных. Смысл их тот же – предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочее) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка – это заметно упрощает работу наборщика, снижает его утомление и предотвращает появление печатных ошибок.

3.4 Отчеты

По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на принтер. В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов.

3.5 Страницы

Это специальные объекты баз данных, реализованных в последних версиях СУБД Microsoft Access (начиная с Access 2000). Правда, более корректно их называть страницами доступа к данным. Физически это особый объект, выполненный в коде HTML, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является базой данной, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и базой данных, размещенной на сервере. Эта база данных не обязательно должна быть базой данных Microsoft Access. Страницы доступа, созданные средствами Microsoft Access, позволяют работать также с базами данных Microsoft SQL Server.

3.6 Макросы и модули

Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с СУБД, так и для создания новых функций путем программирования. В СУБД Microsoft Access макросы состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. Модули создаются средствами внешнего языка программирования, в данном случае языка Visual Basic for Applications. Это одно из средств, с помощью которых разработчик базы может заложить в нее

нестандартные функциональные возможности, удовлетворить специфическое требование заказчика, повысить быстродействие системы управления, а также уровень ее защищенности.

4 ПОНЯТИЕ РЕЛЯЦИОННОЙ МОДЕЛИ И НОРМАЛИЗАЦИЯ

4.1 Реляционная модель

Реляционная модель базы данных представляет собой множество взаимосвязанных двумерных таблиц – реляционных таблиц, в каждой из которых содержатся сведения об одной сущности автоматизированной предметной области.

Каждая строка в таблице содержит данные, относящиеся к некоторой вещи или какой-то её части и называется записью. Каждый столбец таблицы описывает какой-либо атрибут этой вещи. Например, атрибут сущности – город, экземпляр сущности – Ташкент.

4.2 Нормализация

Нормализация – это удаление из таблиц повторяющихся данных путем их переноса в новые таблицы, экземпляры сущности которой не должны содержать повторяющихся значений. Минимальное дублирование данных в реляционной БД обеспечивают высокую эффективность поддержания БД в актуальном и непротиворечивом состоянии, однократный ввод и корректировку данных.

4.3 Ключи

Ключ – это группа из одного или более атрибутов, которая уникальным образом идентифицирует строку. По значению ключа таблицы отыскивается единственная запись в таблице. Ключ может состоять из одного или нескольких атрибутов. Значение уникального ключа не может повторяться в нескольких записях.

4.4 Связи

Между полями, а так же между записями могут быть установлены различные логические связи. Логические связи дают возможность объединять

данные из различных таблиц. Связь каждой пары таблиц обеспечивается одинаковыми полями в них — ключом связи.

Типы реляционных связей:

один к одному — это таблицы, содержащие какие-либо дополнительные сведения об отчётах первой таблицы;

один ко многим — каждая запись главной таблицы может быть поставлена в соответствии с одной или несколькими записей подчинённой таблицы.

5 ПОСТРОЕНИЕ БАЗЫ ДАННЫХ О СТУДЕНТАХ

Для того чтобы лучше понять Access построим свою Базу данных. Сначала определимся, какую информацию будет содержать наша База данных. База данных может содержать информацию о чем угодно: о зарплате, о торговых поставках, о больных в поликлинике, о студентах ВУЗа, о друзьях, о книгах в вашей библиотеке. Мы построим простую - о студентах какого-нибудь ВУЗа. Эта База данных будет состоять из четырёх взаимосвязанных таблиц:

1. Список студентов, который будут содержать такую информацию:

- номер студента.
- имя.
- фамилия.
- специальность.
- имя руководителя.

2. Список дисциплин, который будут содержать такую информацию:

- номер дисциплины.
- название.
- количество часов.

3. Список руководителей, который будут содержать такую информацию:

- имя руководителя.
- телефон руководителя.
- кафедра.

4. Учебный план, который будет содержать такую информацию:

- номер студента.
- номер дисциплины.
- курс.

Графически это может быть представлено в виде четырех таблиц:

Таблица 1. Руководитель

имя руководителя	Телефон	кафедра
Вячеслав	150-26-95	физика
Виктор	385-62-84	черчение
Галина	115-92-60	основы права
Тимур	286-40-62	информатика
Хабиба	120-50-89	математика

Таблица 2. Студент

номер студента	имя	фамилия	специальность	имя руководителя
50	Ренат	Хабибуллин	информационные системы	Вячеслав
78	Римма	Чурикова	бухгалтерский учёт	Виктор
220	Амир	Валиев	маркетинг	Галина
324	Артур	Коновалов	маркетинг	Тимур
460	Женя	Чиркин	информационные системы	Хабиба

Таблица 3. Дисциплина

номер дисциплины	название	количество часов
1	физика	60
2	математика	80
3	черчение	100
4	основы права	50
5	информатика	70

Таблица 4. Учебный план

номер студента	номер дисциплины	курс
50	1	4
78	2	2
220	3	3
324	4	3
460	5	1

Итак, начнем.

Запускаем Access и открываем Базу данных, назовем ее **Студенты**. Сначала нам надо построить три, выше описанных таблицы. Для этого нажимаем кнопки: **Создать – Новая БД** и начнем конструировать первую таблицу. Вводим имена полей и их типы: **Текстовый** или **Числовой**. Затем установим первичный ключ к полю номер студента, но это необязательно. Access при сохранении таблицы спросит: "Создать первичный ключ?" если Вы

выберите **Да**, то он создаст новое поле с именем **Код** и типом **Счётчик**. (Структура таблицы **студент** указана в приложении стр. 1, рис. 1)

Дальше нам нужно установить связи между таблицами, для этого в меню **Сервис** выберите опцию **Схема данных** и установите связи между полями. Для установления связи выберите нужное вам поле, укажите на него мышкой и, удерживая левую кнопку мыши, покажите на поле, с которым Вы хотите установить связь. Но здесь есть небольшое ограничение: поля, между которыми Вы хотите установить связь должны быть одного типа. В данном случае мы установили связь один ко многим. (Смотрите приложение стр. 1, рис. 2)

Эти связи нам нужны для облегчения поиска и выборки информации. Например, при просмотре списка дисциплин Access автоматически устанавливает указатель (т.е. делает запись текущей) на первую запись в таблице **учебный план**, содержащую в поле номер дисциплины такое же число, что и в текущем поле номер дисциплины таблицы **дисциплина**.

Теперь мы можем заполнить какой-либо информацией эти четыре таблицы. Для этого в окне **Базы Данных** нажмите кнопку **Открыть** или щёлкните два раза левой кнопкой мыши, и начинайте заполнять. (Пример заполненной таблицы **студент** указан в приложение стр. 2, рис. 3)

Как вы понимаете это минимальный набор информации по этой теме, но нам и его хватит.

И так на данном этапе мы имеем базу данных, состоящую из четырёх связанных таблиц и содержащую некоторую информацию. Дальше нам надо как-то работать с этой базой и в первую очередь нам надо производить поиск и выборку информации. Для этого существуют запросы.

Построим несколько запросов.

1. Отсортируем таблицу **студент** в алфавитном порядке по полю фамилия. Для этого сделаем QBE-запрос такого вида:

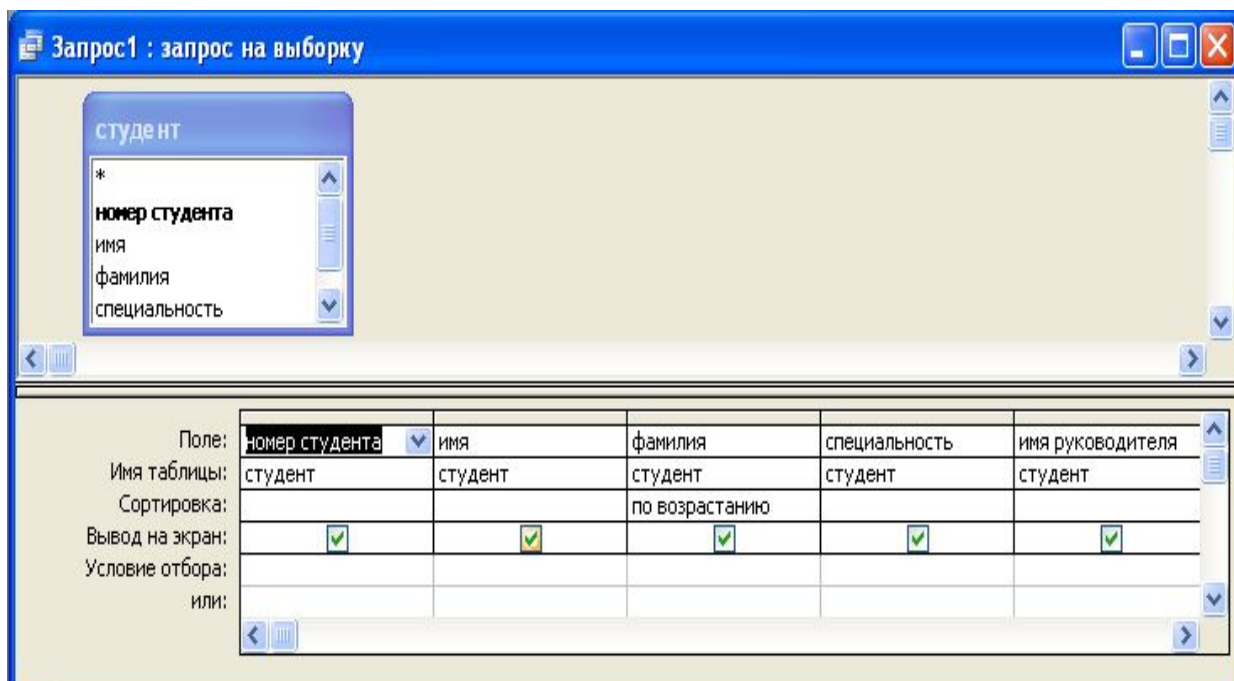


Рис 1.

И если мы посмотрим Запрос 1, то увидим следующую таблицу:

Запрос1 : запрос на выборку

	номер студента	имя	фамилия	специальность	имя руководителя
▶	220	Амир	Валиев	маркетинг	Галина
	324	Артур	Коновалов	маркетинг	Тимур
	50	Ренат	Хабибуллин	информационные системы	Виктор
	460	Женя	Чиркин	информационные системы	Хабиба
	78	Римма	Чурикова	бухгалтерский учёт	Вячеслав
*	0				

Запись: 1 из 5

Рис 2.

2. Далее нам, например, может понадобиться выбрать из списка студентов всех обучающихся на первом курсе. Для этого существуют критерии выборки. Для того чтобы осуществить нужную нам выборку нам надо в QBE-запросе в поле Условие отбора столбца курс поставить цифру '3' (это значит, что из таблицы учебный план будут выбраны все записи, у которых в поле столбца курс стоит '3'). И если мы посмотрим Запрос 2, то увидим следующую таблицу:

Запрос2 : запрос на выборку

	номер студента	номер дисциплины	курс
▶	220	3	3
	324	4	3
*	0	0	0

Запись: 1 из 2

Рис 3.

3. Так как у нас есть несколько связанных таблиц, то мы можем сделать запрос, основанный на нескольких таблицах. Например, из имеющихся у нас таблиц сделаем такую выборку: фамилия студента, имя руководителя, курс, специальность, название дисциплины, количество часов. Для этого подготовим следующий QBE-запрос:

Запрос3 : запрос на выборку

студент: * номер студента, имя, фамилия, специальность, имя руководителя

учебный план: * номер студента, номер дисциплины, курс

дисциплина: * номер дисциплины, название, количество часов

руководитель: * имя руководителя, телефон, кафедра

Поле: фамилия, имя руководителя, курс, специальность, название, количество часов
Имя таблицы: студент, руководитель, учебный план, студент, дисциплина, дисциплина
Сортировка: по возрастанию
Вывод на экран: ☒ ☒ ☒ ☒ ☒ ☒
Условие отбора:
или:

Рис 4.

И если мы просмотрим Запрос 3, то увидим следующую таблицу:

Запрос3 : запрос на выборку

	фамилия	имя руководителя	курс	специальность	название	количество часов
▶	Валиев	Галина	3	маркетинг	черчение	100
	Коновалов	Тимур	3	маркетинг	основы права	50
	Хабибуллин	Виктор	4	информационные системы	физика	60
	Чиркин	Хабиба	1	информационные системы	информатика	70
	Чурикова	Вячеслав	2	бухгалтерский учёт	математика	80
*						

Запись: 1 из 5

Рис5.

Таким образом, вы видите, что Access позволяет осуществлять различную обработку информации.

Но что Вы будете делать, если в Вашу Базу данных ввод информации будет осуществляться неквалифицированным пользователем. Чтобы такие люди могли спокойно работать с Вашей Базой и не видели того, что им ненужно видеть, Access предусматривает такую вещь как **Форма**. Построим две формы.

Из предусмотренных способов создания формы выберем самый простой - это использование Мастера форм. Первую форму сделаем на основе таблиц: студент и руководитель. Укажем поля, которые необходимы для построения нашей формы: имя руководителя, телефон, имя, фамилия, специальность. Выберем: вид представления данных – **руководитель**; внешний вид формы – **табличный**; стиль – **стандартный**. Форма будет выглядеть так:

Рис 6.

Вторая же форма будет на основе таблиц: студент и учебный план. Выберем поля: имя, фамилия, специальность, курс. Теперь сделаем операции, используемые при создании первой формы. Форма будет выглядеть так:

Рис 7.

В этих окнах можно как редактировать, так и вводить новую информацию. Помимо этого, формы можно распечатывать на принтере.

Кроме форм Access предоставляет специальное средство для распечатки выбранной информации - Отчеты. Давайте создадим один, но основанный только на одной таблице студент: Фамилия, имя руководителя, специальность. Далее, сгруппируем по специальности, очередность сортировки сделаем по полю фамилия. Выберем: вид макета для отчёта – ступенчатый; ориентация – книжная; стиль – строгий. Нажмем кнопку **Готово**. (Готовый отчёт Студенты смотрите в приложении стр. 2, рис. 4)

Обработывая, таким образом, информацию мы часто повторяем определенные последовательности операций и для того, чтобы избежать этого, а также повысить эффективность общения с Базой данных и сократить время обработки данных. Access предоставляет **набор Макрокоманд**. В нашей Базе есть одно такое повторяющееся действие: если при каждой загрузке Базы данных мы хотим открывать сразу все наши таблицы, то мы можем записать последовательность макрокоманд делающих это.

Для этого создадим окно проектирования макрокоманд, здесь каждая строка может содержать одну инструкцию в колонке **Макрокоманда** и комментарий к ней в колонке **Примечание**. В каждой строке колонки Макрокоманда есть список макрокоманд, из которого выберем нужную нам макрокоманду: Открытьтаблицу. Повторим это действие четыре раза. После каждого действия будем нажимать **F6**, и указывать имя таблицы.

Окно проектирования макрокоманд будет выглядеть так:

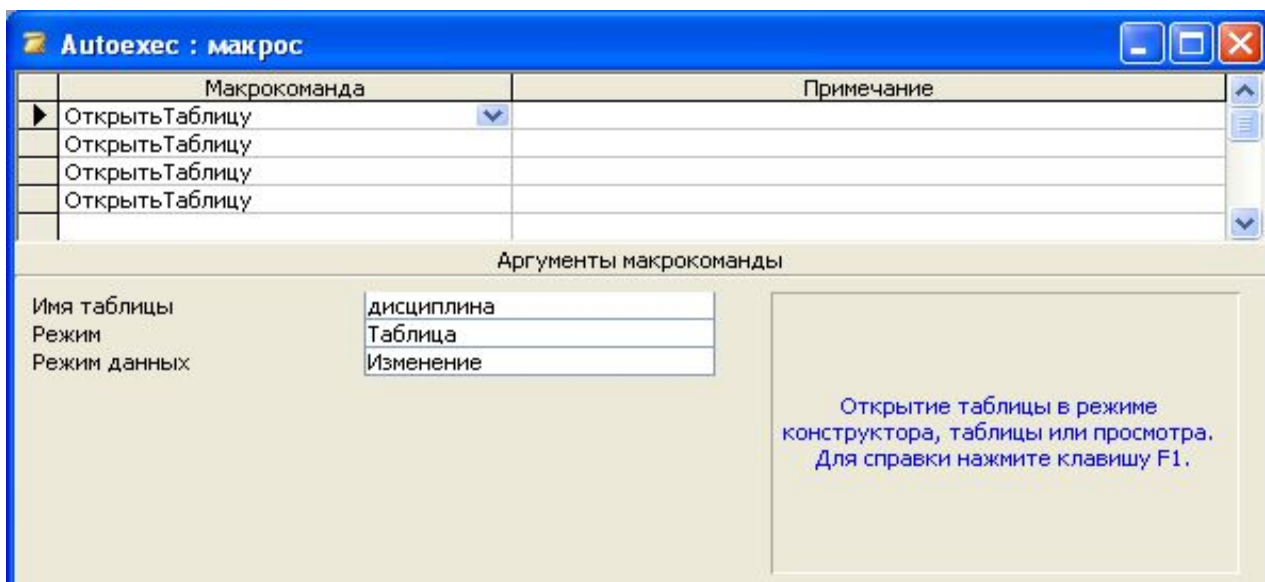


Рис 8.

После закрытия макроса Access спросит сохранить изменения, нажмём Да, введём имя AutoExec. Теперь закроем Access и откроем снова. Мы видим результат:

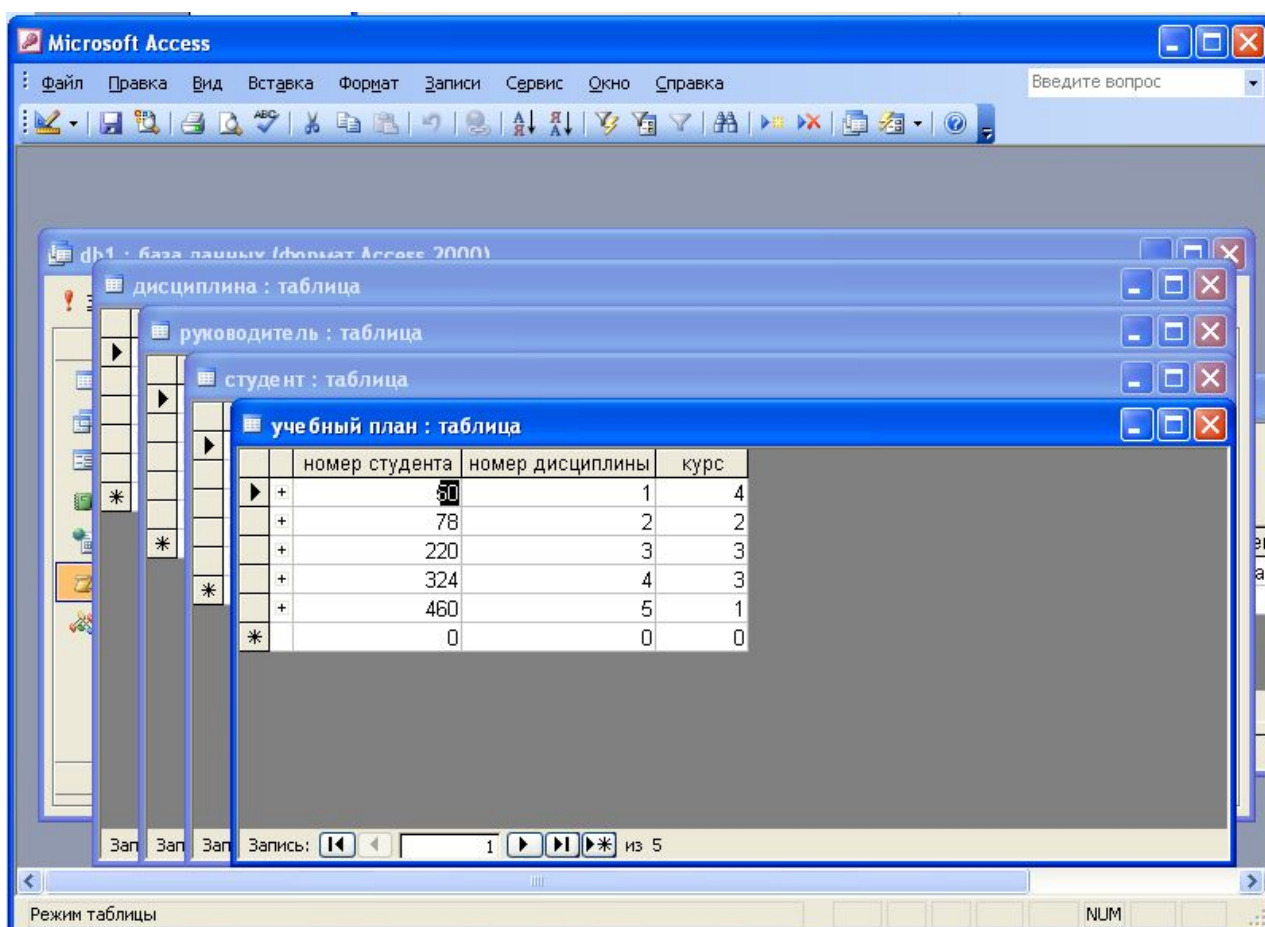


Рис 9.

ЗАКЛЮЧЕНИЕ

Выяснив основную часть данных, с которыми связан студент, можно приступить к созданию структуры базы, то есть структуры ее основных таблиц.

1. Работа начинается с составления генерального списка полей – он может насчитывать десятки и даже сотни позиций.

2. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

3. Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Цель – обеспечить, чтобы ввод данных в одну таблицу производился, по возможности, в рамках одного подразделения, а еще лучше – на одном рабочем месте.

4. В каждой из таблиц намечают ключевое поле. В качестве такого выбирают поле, данные в котором повторяться не могут.

5. С помощью кнопки **схема данных** устанавливаются связи между таблицами.

Для создания нашей Базы Данных мы использовали самый простой и достаточный набор операций, который позволил нам создать:

- Четыре таблицы;
- Три запроса;
- Три формы;
- Один отчет;
- Один макрос.

Microsoft Access – это одна из понятных программ в кругу пользователей, позволяющая удобно работать в Базе Данных. Он имеет очень удобный и понятный интерфейс, очень многие операции у него автоматизированы с помощью различных Конструкторов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ACCESS 2003. Самоучитель с примерами / Ганчаров А. Ю. – М.: КУДИЦ-ОБРАЗ, 2004. - 272 с.
2. Microsoft Access для пользователя / Бемер С., Фратер Г.: Микап, Москва 1994.
3. Информатика. Базовый курс / Симонович С.В. и др. - СПб: Издательство «Питер», 2000. – 640 с.
4. Информатика. Учебное пособие / Под ред. В.Г. Кирия. – Иркутск: ИрГТУ, 1998 часть 2. – 382 с.
5. Информатика. Учебное пособие / Ломтадзе В.В., Шишкина Л.П. – Иркутск: ИрГТУ, 1999. – 116 с.
6. Программирование Access 2002 в примерах / Виллариал Б.: Пер. с англ. - М.: КУДИЦ-ОБРАЗ, 2003. - 496 с.
7. Теория и практика построения баз данных. 8-е издание / Д. Крёнке. – СПб.: Питер, 2003. – 800 с.
8. Освой самостоятельно программирование для Microsoft Access 2002 за 24 часа / Киммел. Пол: Пер. с англ. — М.: Издательский дом "Вильямс", 2003. — 480 с.