

Багц дагуу

БАЗЫ ДАННЬХ



1. Что такое базы данных?

1. Что такое базы данных?

2. Что такое СУБД?

2. Что такое СУБД?

3. Что такое банк данных?

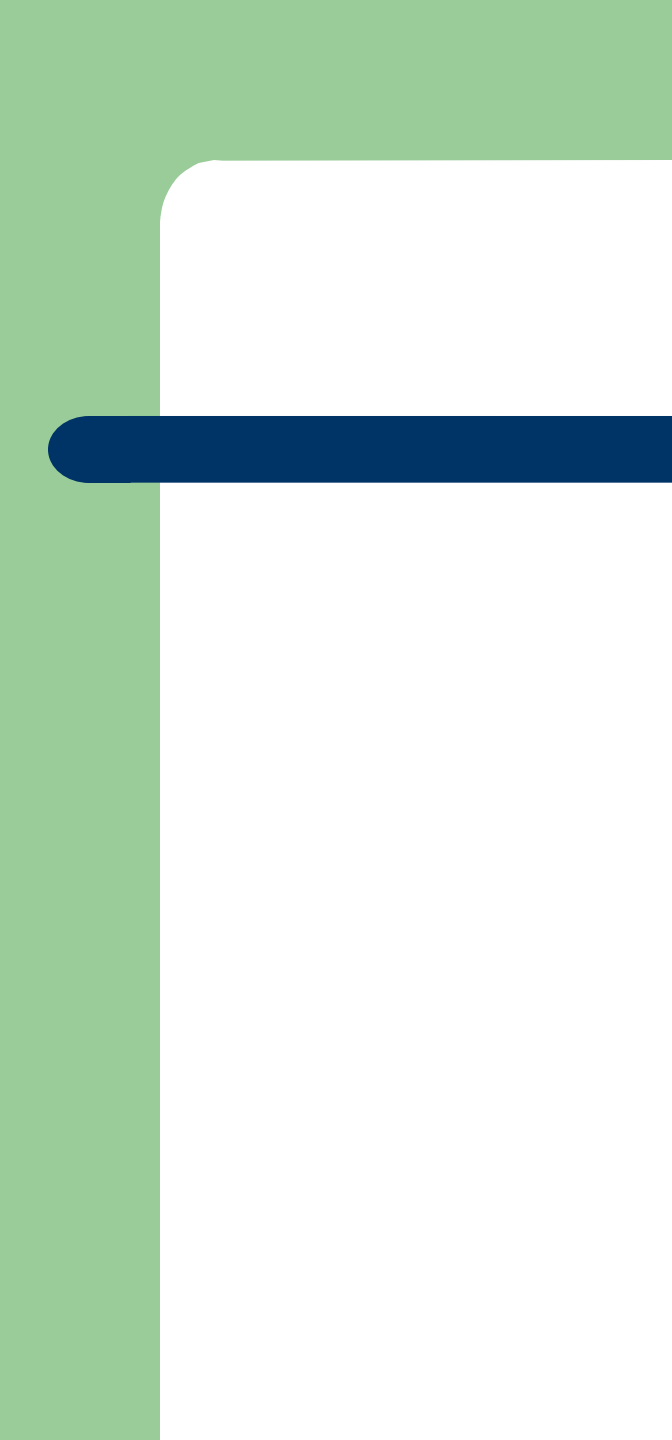
3. Что такое банк данных?

4. Какие операции можно производить

4. Какие операции можно производить

с данными, хранящимися в базе данных?

с данными, хранящимися в базе данных?



❖ **База данных – организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения.**

Базами данных (БД) называют электронные хранилища информации, доступ к которым осуществляется при помощи одного или нескольких компьютеров. База данных состоит из таблиц. Таблицы состоят из полей (столбцов) и записей (строк). Разработчик базы данных определяет структуру БД, т. е. создает поля (задает имя, определяет тип и свойства полей), а пользователь наполняет ее, т. е. вводит, изменяет или удаляет записи.

Примеры: - база данных книжного фонда библиотеки;
- база данных кадрового состава учреждения;
- база данных законодательных актов в области уголовного права;
- база данных современной эстрадной песни.

Как же организована база данных?

Как же организована база данных?

В ней содержатся сведения о большом количестве однотипных объектов.

Каждая база данных предназначена для решения своего класса задач.

Одна нужна, чтобы разыскать книгу по каталогу, другая – для облегчения работы директору образовательного учреждения.

Сейчас в мире созданы сотни тысяч баз данных. Они используются в библиотеках и больницах, в гидрометцентрах и на заводах, в магазинах и планирующих организациях, в банках и частных фирмах.

Для хранения баз данных может использоваться как один компьютер, так и множество взаимосвязанных компьютеров.

Если различные части одной базы данных хранятся на множестве компьютеров, объединённых между собой сетью, то такая БД называется распределённой базой данных.

Информацию в Internet, объединённую паутиной W.W.W., можно рассматривать как распределённую БД.

Распределённые БД создаются также и в локальных сетях.

Информация в базах данных может быть организована по-разному. Чаще всего используется табличный способ.

Базы данных с табличной формой организации называются Реляционными БД.

В чём же их преимущество?

Главное достоинство таблиц в их понятности.

Например: -расписание занятий;

-ведомость с оценками;

-расписание поездов;

-таблица футбольного чемпионата;

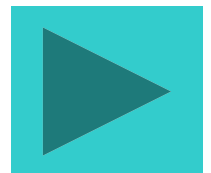
- журнал учителя.

Реляционные базы данных содержат прямоугольные таблицы.

Строка таблицы называется записью, столбец – полем.

	<i>поле1</i>	<i>поле2</i>	<i>поле3</i>	<i>поле4</i>	<i>поле5</i>
<i>запись1</i>
<i>запись2</i>
...					

Таблица имеет главный ключ, отличающий записи друг от друга. Ключом может быть одно поле (простой ключ) или несколько полей (составной ключ).



Одна запись содержит информацию об одном объекте той реальной системы, модель которой представлена в таблице.

Например, в БД – «Домашняя библиотека»- отдельный объект – это книга.

Записи различаются значениями ключей.



Поля – это различные характеристики (иногда , говорят – атрибуты) объекта. Значения полей в одной строчке относятся к одному объекту.

Разные поля отличаются именами.

Важное свойство поля: тип поля – тип определяет множество значений, которые может принимать данное поле в различных записях.

В реляционных базах данных используются четыре основных типа

полей:

- числовой;**
- символьный;**
- дата;**
- логический.**

Значения полей – это некоторые величины определённых типов. От типа величины зависят те действия, которые можно с ней производить.

**Например, с числовыми величинами можно выполнять арифметические операции, а с символьными и логическими –
нельзя.**



Главным ключом в базах данных называют поле (или совокупность полей), значение которого не повторяется у разных записей.

<i>Город</i>	<i>Номер шк.</i>	<i>Директор</i>	<i>Адрес</i>	<i>Телефон</i>
<i>Крюков</i>	<i>1</i>	<i>Иванов А.П.</i>	<i>Пушкина 5</i>	<i>12-35</i>
<i>Шадринск</i>	<i>1</i>	<i>Строев С.С.</i>	<i>Лесная 14</i>	<i>4-23-11</i>
<i>Шадринск</i>	<i>2</i>	<i>Иванов А.П.</i>	<i>Мира 34</i>	<i>4-33-24</i>
<i>...</i>	<i>...</i>	<i>...</i>	<i>...</i>	<i>...</i>

**В этой таблице 2 поля образуют составной ключ (у разных записей не могут совпасть одновременно два поля)
: ГОРОД - НОМЕР ШКОЛЫ.**

Составной ключ может состоять и более чем из двух полей.



В каждой таблице БД должен существовать первичный ключ – одно или несколько полей, однозначно определяющие каждую запись таблицы. Значение первичного ключа должно быть уникальным, то есть в таблице не должно быть двух или более записей с одинаковым значением первичного ключа. Например, если есть таблица по накладным какого-то склада и нумерация накладных каждый месяц начинается с 1, то первичным ключом могут выступать набор из двух полей: Номер и Дата накладной. Потому как могут быть записи с одинаковым номером накладной или с одинаковой датой, но не может быть записи в которой будут одинаковы и номер, и дата.



Системы Управления Базами Данных (СУБД) – это программные средства, предназначенные для создания, наполнения и удаления баз данных. По назначению СУБД подразделяются на три вида: Промышленные универсального назначения, Промышленные специализированные и разрабатываемые под конкретного заказчика. Универсальные рассчитаны «на все случаи жизни» и, как следствие, либо очень сложны в использовании и требуют от пользователя специальных знаний, либо просты, но ограничены в возможностях. Примером универсальных СУБД могут служить Access, FoxPro, Oracle, DB2. Специализированные направлены на выполнение узких задач и потому создаются так, чтобы они были просты в использовании для профессионалов в своей области. Примером таких СУБД могут служить различные бухгалтерские или складские программы (БЭСТ, 1С Предприятие, Правовая система Гарант). СУБД, разрабатываемые под конкретного заказчика, максимально учитывают нужды потребителя, его ситуацию и не требуют дополнительных знаний от пользователя. Но они весьма дороги и требуют времени для создания, отладки и внедрения, тогда как Универсальные и Специализированные сравнительно дешевы и вводятся в эксплуатацию за сравнительно короткий срок (от недели до месяца).

Системы управления базами данных (СУБД) являются едва ли не самым распространенным видом программного обеспечения. СУБД имеют более чем тридцатилетнюю историю развития с сохранением преемственности и устойчивых традиций. Идеологическая ценность СУБД объясняется тем, что в основе программ такого рода лежит концепция модели данных, то есть некоторой абстракции представления данных. В большинстве случаев предполагается, что данные представлены в виде файлов, состоящих из записей. Структура всех записей в файлах одинакова, а количество записей в файле является переменным. Элементы данных, из которых состоит каждая запись, называются полями. Поскольку во всех записях имеются одни и те же поля (с разными значениями), полям удобно давать уникальные имена. Многие практически важные случаи хорошо укладываются в такое представление данных. Например, в отделе кадров информация о сотрудниках имеют такую природу. Сотрудников принимают на работу и увольняют, но форма личного листа по учету кадров остается неизменной для каждого сотрудника.

Товарно-материальные ценности приходят и уходят, но форма инвентарной карточки остается неизменной. Число примеров без труда можно множить. Ясно, что СУБД является адекватным средством во всех случаях, когда исходную информацию можно представить в виде таблицы постоянной структуры, но неопределенной длины или в виде картотеки, содержащей неопределенное количество карточек постоянной структуры.

Все СУБД поддерживают в той или иной форме четыре основных операции:

добавить в базу данных одну или несколько записей;

удалить из базы данных одну или несколько записей;

найти в базе данных одну или несколько записей,

удовлетворяющих заданному условию;

обновить в базе данных значение некоторых полей.

Большинство СУБД поддерживают, кроме того, механизм связей между различными файлами, входящих в базу. Например, связь может установиться явным образом, когда значением некоторых полей является ссылка на другой файл, такие СУБД называются сетевыми, или же связь может установиться неявным образом, например, по совпадению значений полей в различных файлах. Такие СУБД называются реляционными.

MS Access является СУБД реляционного типа, в которой разумно сбалансированы все средства и возможности, типичных для современных СУБД. Реляционная база упрощает поиск, анализ, поддержку и защиту данных, поскольку они сохраняются в одном месте. Access в переводе с английского означает «доступ». MS Access — это функционально полная реляционная СУБД. Кроме того, MS Access одна из самых мощных, гибких и простых в использовании СУБД. В ней можно создавать большинство приложений, не написав ни единой строки программы, но если нужно создать нечто очень сложное, то на этот случай MS Access предоставляет мощный язык программирования — Visual Basic Application.

Популярность СУБД Microsoft Access обусловлена следующими причинами:

доступность в изучении и понятность позволяют Access являться одной из лучших систем быстрого создания приложений управления базами данных;

СУБД полностью русифицирована;

возможность использования OLE технологии;

интегрированность с пакетами Microsoft Office;

поддержка WWW-идеологии (только в Access 97);

визуальная технология позволяет постоянно видеть результаты своих действий и корректировать их; кроме того, работа с конструктором форм может существенно облегчить дальнейшее изучение таких систем программирования, как Visual Basic или Delphi;

широко и наглядно представлена справочная система;

наличие большого набора «мастеров» по разработке объектов.

ИПС – информационно–поисковые системы.

Различие между СУБД и ИПС весьма условное. Пояснить его можно так. Если речь идёт о каталоге библиотеки, то никому и в голову не придёт разрешить любому посетителю изменять в этом каталоге информацию о книгах, хранящихся в её фондах. А просмотреть то, что необходимо, - пожалуйста. Можно сказать, что специалистам – сотрудникам библиотеки – доступны все возможности работы с базой данных, а посетители работают только с информационно – поисковой системой.

Для того, чтобы заставить ЭВМ найти интересующие нас сведения, нужно составить запрос. Правила записи запросов для каждой ИПС свои. Эти правила устанавливаются теми, кто создаёт ИПС. Обычно самые распространённые запросы к базе данных уже заранее составлены. Чтобы получить информацию по такому запросу, надо просто выбрать соответствующий пункт в меню, расположенном на экране. Более того, в большинстве персональных баз данных вообще все запросы заранее составлены, и работа с ними доступна даже человеку, далёкому от компьютерных премудростей.



Проектирование баз данных. Нормализация

Процесс проектирования БД в немалой степени зависит от опыта и интуиции разработчика, т. е. является творческим. Но как быть новичку у которого нет опыта создания баз данных? Кроме того, базы данных хранят огромное количество информации, поэтому могут занимать много места в памяти компьютера, в связи с чем возникает необходимость их оптимизации. Для этих целей были разработаны определенные правила создания структуры БД – нормальные формы. Процесс приведения БД к нормальной форме называется нормализацией. Всего существует семь нормальных форм, но мы рассмотрим только три из них.

Первая нормальная форма (1НФ) требует, чтобы каждое поле таблицы БД было неделимым и не содержало повторяющихся групп. Неделимость поля означает, что содержащаяся в нем информация не должна делиться на более мелкие.

Пример1.1: поле **ФИО** можно разделить на три отдельных поля **Фамилия, Имя, Отчество**.

Пример1.2: нельзя использовать поле **Адрес**, нужно использовать набор полей: **Улица, Номер дома, Квартира, Город**.

Для чего это делается? Предположим в вашей БД существует поле **Адрес**, тогда вам будет очень трудно найти всех людей, живущих на определенной улице. Вам придется писать специальную конструкцию на языке запросов для анализа каждой записи, что чрезвычайно замедлит поиск. При разделении поля вы просто в одно нажатие включаете фильтр по полю и получаете желаемые записи.

Повторяющимися являются поля, содержащие одинаковые по смыслу значения. Пример 1.3: Пусть имеется некое предприятие, продающее крупы. И для них была создана база данных содержащая, помимо данных о накладной и клиенте еще и такие поля: **Сахар, Соль, Мука, Гречка, Рис, Пшено** – в которых хранится количество отпущенного товара. На первый взгляд все хорошо, но каждое поле базы данных вне зависимости от того содержит оно информацию или нет все равно занимает место в памяти, то есть размер, занимаемый пустой записью и полностью заполненной записью одинаков. Предположим, что дела у компании пошли хорошо и она смогла расшириться и стала продавать еще и молокопродукты. Если оставить структуру не тронутой, то в таблицу необходимо добавить новые поля, причем уже имеющиеся записи будут изменены. А если далее число продуктов, продаваемых фирмой превысит хотя бы сотню, то будет ли удобно ориентироваться в такой таблице? В подобной ситуации следует заметить, что все приведенные поля суть наименование товара и хранят они лишь количество товара, поэтому достаточно использовать всего два поля: **Товар, Количество**.

Пример 1.4. Приведение таблицы к 1НФ.

Неприведенная

<i>ФИО</i>	<i>Тетради</i>	<i>Карандаши</i>	<i>Ручки</i>	<i>Ластики</i>
<i>Иванов Иван Иванович</i>		<i>15</i>		
<i>Петров Петр Петрович</i>	<i>10</i>			

Приведенная к 1НФ

<i>Фамилия</i>	<i>Имя</i>	<i>Отчество</i>	<i>Канц. тов.</i>	<i>Кол-во</i>
<i>Иванов</i>	<i>Иван</i>	<i>Иванович</i>	<i>Карандаши</i>	<i>15</i>
<i>Петров</i>	<i>Петр</i>	<i>Петрович</i>	<i>Тетради</i>	<i>10</i>

Вторая нормальная форма (2НФ) требует, чтобы все поля таблицы зависели от первичного ключа, то есть чтобы первичный ключ однозначно определял запись и не был избыточен. Те поля, которые зависят только от части первичного ключа должны быть выделены в отдельные таблицы. Предположим у нас есть таблица для отпуска товара по накладным. Эту таблицу мы должны привести ко второй нормальной форме. Рассмотрим это только схематично:



Первичным ключом здесь являются поля Дата и Номер. Поля: Счет, Город и Телефон – зависят только от поля Покупатель, но не от первичного ключа, поэтому их можно выделить в отдельную таблицу под названием Клиенты. Таким образом, мы привели БД Накладные ко второй нормальной форме.

Третья нормальная форма (3НФ) требует, чтобы в таблице не имелось транзитивных зависимостей между неключевыми полями, т. е. чтобы значение любого поля, не входящего в первичный ключ, не зависело от значения другого поля, также не входящего в первичный ключ.

Типичным примером может служить связка полей: Цена за единицу, Количество, К оплате. Дело в том, что значение поля К оплате зависит от полей Цена за единицу и Количество. Значение поля К оплате можно вычислять в процессе создания отчета или запроса, но хранить его в Базе данных вовсе не обязательно.

Нормализация таблиц позволяет устранить из них избыточную информацию, что приводит к уменьшению размера базы данных. В тоже время чрезмерное увлечение нормализацией может привести к тому, что поиск по БД будет идти очень медленно, поскольку СУБД должна будет считывать данные из нескольких таблиц, тратя время на их открытие, просмотр и закрытие. Поэтому вы всегда должны определить для себя что является наиболее важным: размер базы или скорость поиска. Следует отметить, что первые две нормальные формы применяются очень часто, третья чуть реже. При грамотном использовании нормализации разработчиком, пользователь получает логически выстроенную, удобную в использовании и компактную базу данных.

Создание базы данных. Ввод и форматирование данных

- 1. Загрузите систему WINDOWS. Загрузите СУБД ACCESS. Появятся выплывающие карточки-подсказки. Сверните их. Если вам понадобится подсказка, то в любой момент вы можете вызвать карточки в меню Справка или соответствующей кнопкой на панели инструментов.**
- 2. Сначала нужно создать новую базу данных.**
- 3. Выполним следующую последовательность действий: в меню Файл выберем команду Создать. Имя файла: skaz.mdb. ОК. Перед вами появилось диалоговое окно «База данных».**
- 4. Внимательно прочитайте назначение кнопок на панели инструментов, медленно перемещая курсор мыши по кнопкам.**
- 5. После этого создайте таблицу, выполнив следующую последовательность действий: Таблица/Создать/Новая таблица.**

Создание таблицы, то есть определение входящих в таблицу полей, производится заполнением специальной таблицы:

Поле	Тип данных	Описание

6. Заполните такую таблицу, внося в нее следующие данные:

Поле	Тип данных	Описание
№	Счетчик	
Персонаж	Текстовый	
Профессия	Текстовый	
Особые приметы	Текстовый	
Герой	Логический	Положительный или отрицательный герой

7. Поле № не обязательное, мы его вводим для того, чтобы определить ключевое поле, так как любая таблица должна иметь ключ.

8. Созданную таблицу нужно сохранить, дав ей имя с помощью команд: Файл/Сохранить как..., Имя таблицы: «Персонаж», ОК.

9. Введите информацию в таблицу Таблица/«Персонаж» /Открыть и обычным образом введите данные, например такие:

№	Персонаж	Профессия	особые приметы	герой
1	Буратино	деревянный человечек	длинный нос	Да
2	Папа Карло	Шарманщик		Да
3	Карабас Барабас	директор кукольного театра	длинная борода, достающая до пола	Нет
4	Лиса Алиса	Мошенница	хромая на одну ногу	Нет
5	Кот Базилио	Мошенник	слепой на оба глаза	Нет
6	Мальвина	артистка театра	девочка с голубыми волосами	Да
7	Дуремар	Фармацевт	характерный запах тины	Нет
8	Тортилла	хранительница золотого ключика	черепаха	Да

10. При помощи мыши выделите:

а) запись 5,

б) запись 3,

в) с третьей по седьмую запись. Отмените выделение.

г) Выделите все записи. Отмените выделение.

д) Выделите поле «Персонаж».

е) Выделите одновременно поля: «Профессия», «Особые приметы» и «Герой», отмените выделение.

ж) Выделите все поля. Это можно сделать при помощи мыши или в меню Правка выбрать команду Выделить все записи.

11. Отмените выделение.

12. Выделите:

а) В поле «Особые приметы» отметьте шестую запись.

б) В поле «Персонаж» выделите с четвертой по шестую запись.

в) Не отпуская кнопку мыши, отметьте эти же записи в полях «Особые приметы» и «Герой».

13. Отмените выделение.

14. Выделите всю таблицу.

15. Отмените выделение.

16. Измените ширину каждого столбца, так чтобы ширина колонок была минимальной, но был виден весь текст.

Это можно сделать при помощи мыши, раздвинув столбцы или следующим образом.

Выделите нужный столбец и нажмите правую кнопку мыши, в контекстном меню выберите команду «Ширина столбца»; в открывшемся окне нажмите кнопку По ширине данных.

Проделайте такую же работу со всеми полями.

Высоту строки можно изменить аналогичным образом с помощью мыши или в меню Формат командой Высота строки. Причем достаточно отредактировать одну строку, высота остальных строк изменится автоматически.

17. Любым способом измените высоту строки и сделайте ее равной 30.

18. Измените шрифт таблицы на Arial Cyr, размер шрифта 14, полужирный.

Изменить шрифт можно так: вывести указатель мыши за пределы таблицы и нажать левую кнопку мыши, в контекстном меню выбрать Шрифт или в меню Правка на панели инструментов выбором команды Шрифт.

19. Измените шрифт текста на Times New Roman Cyr, размер шрифта 10.

20. Измените ширину полей.

а) Сделайте столбец «Персонаж» шириной 20.

б) Столбец «Особые приметы» шириной 25.

Вы видите, что текст в этих полях напечатался в две строки.

21. Подгоните ширину столбцов так, чтобы текст вмещался полностью.

22. Выполните сортировку таблицы по полю «Персонаж» в порядке, обратном алфавитному.

Это можно сделать так. Выделите поле «Персонаж» и нажмите кнопку Сортировка по убыванию на панели инструментов.

23. Верните таблицу в исходное состояние.

24. Сохраните таблицу «Персонаж».

25. Закройте таблицу «Персонаж».

Занятие 2. Редактирование базы данных

1. Откройте таблицу «Персонаж» и добавьте в конец таблицы следующие записи:

<i>№</i>	<i>Персонаж</i>	<i>Профессия</i>	<i>особые приметы</i>	<i>герой</i>
1	Артемон	Пудель	очень умный пес	Да
2	Трактирщик	хозяин таверны "Три пескаря"	жаден и скуп	Нет

2. Это можно сделать тремя способами:

а) Передвинуть курсор в конец таблицы и ввести новые записи.

б) На панели инструментов нажать кнопку Новая запись.

в) В меню Записи выбрать команду Ввод данных.

3. Скопируйте первую запись на место шестой записи.

4. Удалите пятую запись.

5. Скопируйте первую запись в конец таблицы.

6. Измените профессию Дуремара на продавец пивок.

Это можно сделать так: отметить курсором мыши запись фармацевт, удалить ее в буфер и с клавиатуры ввести продавец пивок. Или следующим способом: открыть меню Правка на панели инструментов, выбрать команду Заменить... На экране появится диалоговое окно замены. Ввести формат замены.

7. Замените во всей таблице Буратино на Пиноккио.

8. Удалите две записи, значение которых в поле «Персонаж» = Пиноккио

9. Вставьте перед четвертой записью следующую запись:

*№ Персонаж Профессия особые приметы герой 1 Пьеро артист
театра грустные глаза Да*

10. Удалите с седьмой по одиннадцатую записи.

11. Удалите поле «Профессия».

Это можно сделать с помощью мыши, соединив границы полей «Персонаж» и «Особые приметы» или в меню Правка командой Удалить поле.

12. Вставьте новое поле в таблицу.

Поле	Тип данных	Описание
Дата	Числовой	Условная дата рождения (целое число)

- 13.** Чтобы вставить новое поле в таблицу, нужно в окне базы данных выбрать корешок Таблица, нажать кнопку Конструктор. В открывшемся окне поставьте курсор на ту строку, перед которой вам необходимо вставить запись. Нажмите кнопку Вставить строку на панели инструментов и введите нужные данные.
- 14.** Самостоятельно заполните новое поле в каждой записи.
- 15.** Во всей таблице замените слог но на слог ни.
Для этого в любой записи таблицы необходимо отметить слог но. Затем в меню Правка выберите команду Заменить, и введите формат замены.
- 16.** Замените в таблице все буквы о на букву а, а буквы а замените на букву о.
- 17.** Замените на слог РА буквосочетание, первая буква которого р, а вторая — любая буква русского алфавита. Это можно сделать следующим образом: в любой записи таблицы отметить необходимое буквосочетание. Например, слог ер. Затем в меню Правка выбрать команду Заменить и ввести формат замены. Заменить ?р на РА.

Поиск данных по условию

Следующее задание, которое мы вам предлагаем, можно выполнить различными способами.

Способ первый. Вы создаете запрос по каждому из вопросов, которые предлагаются, и сохраняете его в виде таблицы. Для создания запроса выполните следующие действия: Запрос — Создать — Новый запрос — Таблица/запрос — Класс — Добавить. — Заккрыть. После этого ввести условия запроса.

Способ второй. Таблица — Класс — Открыть. На панели инструментов нажмите кнопку Изменить фильтр, и с помощью построителя выражений введите условия запроса. Затем на панели инструментов Запроса нажмите кнопку Применить фильтр. В результате на экране появится динамический набор записей, удовлетворяющих введенному условию. Но при дальнейших манипуляциях этот набор не сохранится. Если вы хотите сохранить данный набор записей, то выполните следующие действия. В окне Фильтра откройте меню Файл и выберите команду Сохранить как запрос.

Способ третий. В режиме таблиц выведите курсор мыши за пределы таблицы и нажмите правую мыши. В контекстном меню выберите команду Изменить фильтр. Все остальные действия точно такие же, как во втором способе.

Для того чтобы ответить на все предлагаемые вопросы, необходимо знать операторы задания шаблонов. Приведем несколько примеров.

Примеры использования символов задания шаблонов

Использование символов	Образец	Поиск в Microsoft Access
Вопросительный знак (?); в качестве шаблона для любого символа.	За?ор	Забор Затор
Звездочка (*); в качестве шаблона для любой группы символов.	Д*нь	День Добрый день Длинная тень
Звездочка (*); в качестве шаблона для любой группы начальных символов.	*й	128й Последний
Знак фунта (#); в качестве шаблона для любой цифры.	#-й	5-й 8-й
Восклицательный знак (!) после первой скобки; для поиска символа, который не входит	Иванов [!аы]	Иванову [но не Иванова или Ивановы]

1. В базе данных skaz.mbd откройте новую таблицу «Класс» и внесите в нее следующие поля.

Поле	Тип данных	Описание
№	Счетчик	
Фамилия	Текстовый	
Имя	Текстовый	
Дата	Дата	Дата рождения
Пол (м)	Логический	Пол мужской?
Улица	Текстовый	
Дом	Числовой	
Квартира	Числовой	
Класс	Числовой	
Группа	Текстовый	Группа здоровья по физкультуре
Хобби	Текстовый	
Глаза	Текстовый	Цвет глаз

1. Заполните таблицу.

№	Фамилия	Имя	Дата	пол	Улица	Дом	Квартира	Клас с	Группа	Хобби	Глаза
1	Суханов	Сергей	16.02.81	Да	Чердынск ая	23	74	10	основная	тяжела я атлетик а	зелены е
2	Пирогов	Юрий	5.12.83	Да	Куйбышев а	6	31	8	основная	футбол	голубы е
3	Лебедева	Света	16.06.85	Нет	Пушкина	37	65	6	специальная	вязание	карие
4	Голдобин	Сергей	23.05.88	Да	Леонова	12	10	3	основная	лыжи	голубы е
5	Ельшина	Наташ а	24.05.82	Нет	Чердынск ая	37	48	9	специальная	чтение	серые
6	Суханова	Наташ а	20.12.86	Нет	Ленина	12	22	5	подготовитель ная	шитье	зелены е
7	Петрова	Света	18.04.82	Нет	Пушкина	37	3	9	основная	лыжи	серые
8	Горина	Оля	20.12.84	Нет	Связева	66	99	7	подготовитель ная	аэробик а	карие
9	Попов	Михаи л	7.07.87	Да	Леонова	72	6	4	подготовитель ная		голубы е
10	Сергеев	Саша	30.11.89	Да	Куйбышев а	3	31	2	основная	каратэ	зелены е
11	Павлова	Елена	13.12.85	Нет	Пушкина	5	6	6	основная	аэробик а	карие
12	Емельяно ва	Наташ а	25.05.81	Нет	Попова	40	47	10	основная	шитье	зелены е
13	Евдокимо	Михаи	18.08.80	Да	Чердынск	3	40	7	основная	футбол	зелены

Ответьте на вопросы:

1. Кто учится в 8 классе?
2. Кто родился в 1984 году?
3. Кто живет на улице Пушкина?
4. У кого номер дома меньше 50?
5. У кого мужские имена?
6. Кто родился весной?
7. У кого фамилия начинается на букву С?
8. Кто закончит школу в этом году?
9. Кто не занимается тяжелой атлетикой?
10. У кого сегодня день рождения?
11. У кого номер квартиры меньше 12?
12. Кто не определил свое хобби?
13. Кто родился до 1.01.85?
14. Кто живет на улице Пушкина, Леонова или Свизева?
15. У кого фамилия начинается с букв А-К?
16. Кто является однофамильцами с Сухановым Сергеем?
17. Кто увлекается аэробикой и лыжами?
18. Кто из мальчиков живет на улице Чердынской?
19. У кого имена начинаются на букву С, а фамилия — на букву П?
20. У кого карие глаза и этот человек не умеет вязать?
21. Кто учится в старших классах?
22. У кого из специальной группы женское имя?
23. Кто не живет на улице Ленина и не занимается в подготовительной группе?

24. Кто из мальчиков с голубыми глазами занимается каратэ?
25. У кого номер класса кратен 3, и он живет на улице Пушкина или Ленина?
26. Кто с серыми глазами учится в классе с нечетным номером, или число в дате его рождения больше 15?
27. У кого номер класса простое число, или специальная группа, и его имя начинается на Н?
28. Кто из основной группы не живет на улице Пушкина или Ленина?
29. Кто из девочек, родившихся в мае или в феврале, номер дома больше 36?
30. У кого из основной группы имя начинается на буквы С или М, но фамилия не начинается на букву П?

Занятие 4. Работа с помощью мастера

1. *Создайте в новом файле таблицу «Класс» с помощью Мастера таблиц. Для этого откройте корешок Таблица и выполните следующие действия: Создать — Мастер. На экране появится окно мастера таблиц. Внимательно просмотрите примеры таблиц и полей, предлагаемых мастером.*
2. *Выберите поля, необходимые для нашей таблицы «Класс». Недостающие поля можно будет добавить с помощью Конструктора таблиц.*
3. *Выполните все действия, которые вам предлагает сделать мастер. Дайте новой таблице имя «Класс».*
5. *Заполните таблицу и выполните задания, которые мы вам предлагали на предыдущем занятии.*
6. *С помощью мастера создайте Запрос положительных героев в таблице «Персонаж».*
7. *С помощью мастера создайте Запрос отрицательных героев в этой же таблице.*
8. *С помощью мастера самостоятельно создайте Форму для таблицы «Класс».*
9. *Создайте различные типы форм и дайте им различные имена.*
10. *С помощью мастера создайте Отчет для таблицы «Класс», также попробуйте сделать различные виды отчетов.*

Занятие 5. Работа со связанными таблицами

1. Создадим базу данных, которая будет хранить результаты экзаменов студентов одной группы за один семестр. Можно обойтись и одной таблицей, в которую включить поля «Фамилия», «Имя», «Дисциплина», «Преподаватель», «Оценка». Но тогда очень много данных будет повторяться, т.к. каждый студент сдает несколько экзаменов, и каждый экзамен сдают многие студенты.
2. Поэтому создадим три таблицы — «Студенты», «Предметы», «Оценки» — и заполним их. В первых двух таблицах первое поле имеет тип счетчик, а остальные — текстовый тип. В третьей таблице первое поле имеет тип счетчик, а остальные — числовой тип. Таблица «Предметы»

Код предмета	Название	Преподаватель
1	Программирование	Половина
2	Психология	Сивак
3	Статистика	Полыгалова
4	Информационные системы	Клигман

Код студента	Фамилия	Имя
1	Антонова	Марина
2	Бирих	Яков
3	Волков	Константин
4	Волошина	Светлана
5	Кашина	Наталья
6	Ленц	Алексей
7	Матвеев	Роман
8	Наумова	Татьяна
9	Юдинцева	Оксана

**3. Таблица
«Студенты»**

Код оценки	Код студента	Код предмета	Оцен ка
1	1	1	5

**4. Таблица
«Оценки»**

5. Щелкните на пиктограмму «Схема данных». Вы увидите созданные нами таблицы и связи, которые создал Access. Он автоматически связал поля с одинаковыми именами и совместимыми типами. В других случаях это можно сделать вручную, с помощью мыши — достаточно перетащить одно поле на другое и получить между ними связь.

6. Обратите внимание, что выставлять оценки неудобно. Чтобы выставить К. Волкову «отлично» по численным методам, надо помнить коды студента и предмета. К счастью, в Access можно оперировать с кодами, а информацию выдавать в текстовой форме. Сделаем это для предметов.

7. Выберите в таблице «Оценки» поле «Код предмета» и укажите тип «мастер подстановок». В качестве таблицы источника данных выберите «Предметы» и поля «Код» и «Название». Теперь в режиме таблицы можно выбирать названия из списка.

8. Сделайте то же самое для студентов. А теперь сделаем так, чтобы подстановка выполнялась из двух полей, то есть выводила фамилию и имя. Для этого выберите поле «Код студента», щелкните на вкладке «Подстановка», выберите «Источник строк» и щелкните кнопку с многоточием справа. Откроется построитель запросов, работа с которым не отличается от работы с обычным запросом. Вместо поля «Фамилия» введите строку [Фамилия]&" "&[Имя]. Теперь при подстановке видна и фамилия студента, и его имя.

9. **Добавьте еще двух-трех студентов в таблицу «Студенты».**
Перейдите в таблицу «Оценки». Обратите внимание, что добавленные студенты оказались в конце списка. Чтобы упорядочить список по алфавиту, вызовите построитель запросов для поля «Код студента» и установите сортировку по возрастанию. Сделайте то же самое для поля «Код предмета».
10. **Поле «Код оценки» нам (в отличие от Access) неинтересно. Выделите этот столбец в режиме таблицы и выберите команду Формат/ширина столбца. Установите ее в ноль. Этому же результату можно добиться «перетягиванием» правой границы столбца к левой.**
11. **Обратите внимание, что Access позволяет хранить несколько записей о сдаче одним и тем же студентом одного и того же предмета. Сделаем так, чтобы набор студент-предмет стал уникальным. Для этого в режиме конструктора установим для этих полей опцию «ключевое поле». Это можно сделать через пункт меню Правка или через панель инструментов.**
12. **Изменим подписи к полям «Код студента» и «Код предмета». Для этого выберите строку «Подпись» на вкладке «Общие». Теперь таблица «Оценки» выглядит так:**

**Таблица
«Оценки»**

Студент	Предмет	Оценк а
Антонова Марина	Информационные системы	5
Бирих Яков	Информационные системы	4
Волков Константин	Информационные системы	5
Волошина Светлана	Информационные системы	5
Антонова Марина	Психология	5
Бирих Яков	Психология	5
Волков Константин	Психология	4
Волошина Светлана	Психология	5

Задания

1. Измените ввод в поле Оценка, чтобы вместо цифр выставлялось «отлично», «хорошо» и т. д. Для этого создайте отдельную таблицу и используйте мастер подстановок. При создании таблицы учтите, что существует четыре варианта оценки: «1 — отлично», «2 — хорошо», «3 — удовлетворительно», «4 — неудовлетворительно». Вызовите схему данных и установите связь.



2. Заполните таблицу «Оценки».

3. Рассчитайте средний балл каждого студента. Для этого поместите в конструктор запросов таблицы «Студенты» и «Оценки». Выберите поля «Фамилия», «Имя», а вместо «Оценка» введите «6 - оценка». Это необходимо, так как отличной оценке соответствует 1, хорошей — 2 и т.д. Найдите на панели инструментов кнопку «Групповые операции» и нажмите ее. Примените групповую операцию «среднее» — Avg. Access автоматически присвоило построенному выражению имя «Выражение-1». Измените его на «Средний балл».

4. Выведите список студентов, сдавших экзамены на оценки "хорошо" и "отлично". Для этого проделайте те же операции, что и в предыдущем пункте, но в качестве групповой операции выберите Min — нахождении минимума и задайте условие на отбор — >3. Снимите флажок вывода столбца минимальной оценки на экран.

5. Сохраните результаты работы в файле `students.mdb`.

Занятие 6. Дополнительные возможности Access

1. Откройте базу данных *students.mdb*.
2. Создадим удобную форму для ввода информации. Для этого вызовите мастера форм и задайте следующие поля:
из таблицы «Оценки» — Предмет, Оценка;
из таблицы «Студенты» — Фамилия, Имя;
из таблицы «Предметы» — Преподаватель.
3. Выберите представление данных по студентам, с использованием подчиненной формы.
4. Откройте подчиненную форму и примените сортировку по возрастанию для предметов.

Студенты

Фамилия: Антонова

Имя: Марина

Предмет	Оценка	Преподаватель
Информационные системы	Отлично	Клигман
Программирование	Отлично	Половина
Психология	Отлично	Сивак
Статистика	Отлично	Полыгалова

Запись: 2 из 4

Запись: 1 из 11

5. **Откройте форму студенты в режиме конструктора. Выделите и удалите метку «оценки» (сверху от места внедрения подчиненной формы). Подгоните «вручную» размеры формы.**
6. **Обратите внимание на две навигационные панели внизу формы, — одна из них позволяет просмотреть информацию по студентам, вторая по предметам. Создайте свою навигационную панель. Для этого перейдите в режим конструктора, вызовите панель элементов (воспользуйтесь панелью инструментов или пунктом меню «Вид»). Убедитесь, что включен режим мастеров (кнопка с изображением волшебной палочки вдавлена). Выберите элемент «Кнопка» и поместите его внутрь формы, установите действие «Первая запись», подберите картинку.**
7. **Сделайте еще 4 кнопки — «Предыдущая запись», «Следующая запись», «Последняя запись», «Добавить запись».**
8. **Выделите все 5 кнопок. Это можно сделать двумя способами.**
 - **Щелкните все кнопки, которые хотите выделить, по очереди, удерживая клавишу Shift.**
 - **Используйте «лассо». Нажмите кнопку мыши и, не отпуская ее, растяните рамку вокруг всех кнопок.**

9. Установите выравнивание по нижнему краю, сделайте равным интервал между кнопками, а затем уменьшите его до нуля (пункт меню **Формат**).

10. Щелкните правой кнопкой мыши на квадрат в левом верхнем углу формы. Выберите свойства и установите «поле номера записи» в значение **Нет**.

Студенты

Фамилия: Бирих

Имя: Яков

	Предмет	Оценка	П
▶	Информационные системы	Хорошо	Клигман
▢	Программирование	Хорошо	Половина
▢	Психология	Отлично	Сивак
▢	Статистика	Отлично	Полыгалова
*			

Следующая запись

11. *Сделайте то же самое для подчиненной формы.*
12. *Откройте готовую форму «Студенты».*
13. *Создайте самостоятельно форму, аналогичную экзаменационной ведомости — в верхней ее части должны быть расположены название предмета и фамилия преподавателя, а ниже — список студентов с оценками. Воспользуйтесь мастером, а для окончательной доработки — режимом конструктора.*
14. *Переименуйте созданные формы в «зачетные книжки» и «ведомости».*
15. *Создайте кнопки, которые позволяли бы переходить от одной формы к другой. Выберите вкладку «Макросы», команду ОткрытьФорму и имя из предложенного списка («Ведомости»). Создайте аналогичный макрос для «Зачетных книжек».*
16. *Откройте форму «Ведомости» в режиме конструктора и «перетащите» на нее мышкой макрос, открывающий «Зачетные книжки». Если все сделано верно, кнопка будет создана автоматически. Для редактирования ее внешнего вида, щелкните правой кнопкой и выберите пункт Свойства.*

17. Сделайте то же для другого макроса.

Создайте макрос, который открывает обе формы, и назовите его Autoexes. Сохраните базу, закройте и заново откройте ее. 18. Написанный макрос выполняется автоматически.

Занятие 7. Зачётное задание

1.Создайте базу данных «Книжная энциклопедия», содержащую сведения о произведениях различных писателей. Учтите, что у одного автора может быть много произведений. Продумайте схему данных, создайте таблицы и заполните базу данных следующей информацией:

Автор	Название	Год издания	Худож ?	Издательство
Гамильтон, Э.	Возвращение к звездам	1992	Да	«Селена»
Гамильтон, Э.	Оружие извне	1995	Да	«Армада»
Уильямсон, Д.	Звездный легион	1994	Да	«Армада»
Шарф, Д.	Основы HTML	1997	Нет	«Питер»
Сборник	Лучшие игры для IBM	1994	Нет	«Питер»
Сборник	Java: 1001 совет	1998	Нет	«Попурри»
Пономаренко, С.	Adobe Photoshop 3.0	1995	Нет	BHV
Пономоренко,	Corel Draw 5.0	1995	Нет	BHV

- 2. Добавьте по своему усмотрению еще несколько строк.*
- 3. Выполните следующие задания (создайте запросы):
укажите все нехудожественные произведения, вышедшие после 1995 года;
сколько книг выпущено каждым издательством.*
- 4. Найдите середину интервала, во время которого вышли книги, содержащие в заглавии слова «звезда», «межзвездный» и пр.*
- 5. Создайте простые формы для ввода фамилий авторов и издательств.*
- 6. Создайте формы, которые позволяют просмотреть информацию о книгах по авторам и по издательствам, а также вводить данные по книгам. Используйте знания, полученные на прошлом занятии.
Свяжите эти формы кнопками, вызывающими макросы.*
- 7. Создайте специальную форму, на которой разместите кнопки, вызывающие остальные формы. Создайте макрос ее автозапуска и минимизируйте окно базы данных.*
- 8. Закончите оформление приложения.*

