|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование разделов и тем** | **Количество часов** | | |
| **Всего** | **Теорет.** | **Практ** |
| **1** | **2** | **3** | **4** | **5** |
| 22 | MS Access. Технология хранения и поиска данных. Основные понятия баз данных. Типы данных. | 2 | 2 |  |

**Тема: MS Access. Технология хранения и поиска данных. Основные понятия баз данных. Типы данных.**

Учебные вопросы

1. Понятие базы данных
2. Классификация баз данных
3. Модели баз данных (реляционная, сетевая, иерархическая)
4. Функции СУБД и объекты СУБД Access
5. Типы данных.
6. **Понятие базы данных**

**База данных** — это набор сведений, относящихся к определенной теме или задаче, такой как отслеживание заказов клиентов или хранение коллекции звукозаписей. Если база данных хранится не на компьютере или на компьютере хранятся только ее части, приходится отслеживать сведения из целого ряда других источников, которые пользователь должен скоординировать и организовать самостоятельно.

Другими словами: **Базы** **данных** представляют собой информационные модели, содержащие данные об объектах и их свойствах. Базы данных хранят информацию о группах объектов с одинаковым набором свойств.

Например, база данных «Записная книжка» хранит информацию о людях, каждый из которых имеет фамилию, имя, телефон и т. д. Библиотечный каталог хранит информацию о книгах, каждая из которых имеет название, автора, год издания и т. д.

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

**Структурирование** – это введение соглашений о способах представления данных.

Неструктурированными называют данные, записанные, например, в текстовом файле.

*Пример 1.* На рис. 1 пример неструктурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя отчество и год рождения). Легко убедиться, что сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде, а упорядочить подобную информацию практически не предоставляется реальным.

|  |
| --- |
| Личное дело № 25689, Иванов Иван Иванович, дата рождения 1 января 1980 г.; Л/д №25231, Петрова Мария Васильевна, дата рожд. 8 марта 1975 г., № личн. дела 25549, д.р. 14.02.77 г., Сидоров Василий Петрович. |

Рисунок 1 – Пример неструктурированных данных

Чтобы автоматизировать поиск и систематизировать эти данные, необходимо выбрать определенные соглашения о способах представления данных, т.е. дату рождения нужно записывать одинаково для каждого студента, она должна иметь одинаковую длину и определенное место среди остальной информации. Эти же замечания справедливы и для остальных данных (номер личного дела, фамилия, имя, отчество).

*Пример 2.* После проведения несложной структуризации с информацией, указанной в примере (рис.1), он будет выглядеть так, как показано на рис.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № личного дела | Фамилия | Имя | Отчество | Дата рождения |
| 25689 | Иванов | Иван | Иванович | 01.01.80 |
| 25231 | Петрова | Мария | Васильевна | 08.04.75 |
| 25549 | Сидоров | Василий | Петрович | 14.02.77 |

Рисунок 2 – Пример структурированных данных

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, я также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые *конечными пользователями.*

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария – *системы управления базами данных*.

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

**Система управление базами данных (СУБД)** – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данными, хранимыми в базе.

1. **Классификация баз данных**

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

*Централизованная база* данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

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

По *способу доступа* к данным базы данных разделяются на базы данных с *локальным доступом* и базы данных *с удаленным (сетевым) доступом*.

Системы централизованных баз данных с сетевым доступом предполагают различные архитектуры подобных систем:

* файл-сервер;
* клиент-сервер.

***Файл сервер.*** Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине храниться совместно используемая централизованная БД, Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монопольно. Концепция файл-сервер условно отображена на рис 3.

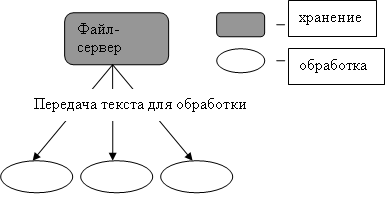


Рисунок 3 – Схема обработки информации в БД по принципу *файл-сервер.*

***Клиент-сервер***. В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL. Концепция клиент-сервер условно изображена нарис. 4.

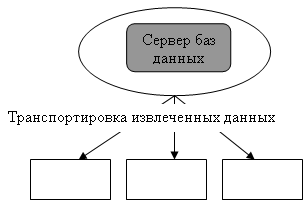


Рисунок 4 – Схема обработки информации в БД по принципу *клиент-сервер.*

Структурные элементы баз данных.

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица) (рис.5).

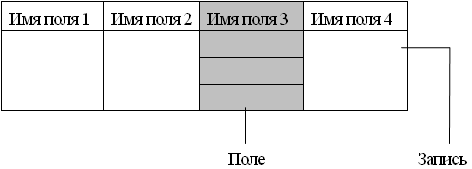


Рисунок 5 – Основные структурные элементы БД

*Поле –* элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие *характеристики:*

*- имя,* например. Фамилия, Имя, Отчество, Дата рождения:

*- тип,* например, символьный, числовой, календарный;

*- длин,* например, 15 байт, причем будет определяться максимально возможным количеством символов;

*- точность* для числовых данных, например два десятичных знака для отображения дробной части числа.

**Запись** – совокупность логически связанных полей. Экземпляр записи – отдельная реализация записи, содержащая конкретные значения ее полей. **Файл (таблица**) – совокупность экземпляров записей одной структуры

Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики, как это покатано на рис. 6.

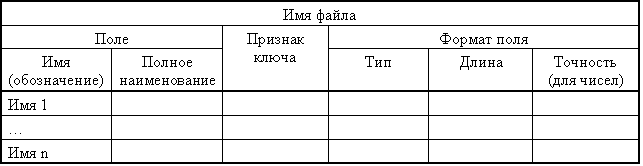


Рисунок 6 – Описание логической структуры записи файла

В структуре записи файла указываются поля, значения которых являются *ключами: первичными* (ПК), которые идентифицируют экземпляр записи, и *вторичными* (ВК), которые выполняют роль поисковых или группировочных признаков (по значению вторичного ключа можно найти несколько записей).

**3. Модели БД**

Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области ивзаимосвязи между ними.

**Модель данных** – совокупность структур данных и операций их обработки.

СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную.

***Реляционные БД***

Это базы данных, данные в которых представлены в виде таблиц, с помощью которых можно описать предметную область. Реляционная БД представляет собой двумерный массив (таблицу).

Термин «реляционный» произошел от англ. слова relation – отношение.

Отношение – математическое понятие, но в терминологии моделей данных отношения удобно изображать в виде таблицы.

Теоретической основой этой модели стала теория отношений американца Чарльза Пирса и немца Эрнеста Шредера. Ими было показано, что множество отношений замкнуто относительно некоторых специальных операций и образует вместе с ними абстрактную алгебру. Американский математик Э.Ф. Кодд в 1970 ᴦ. впервые сформулировал основные понятия и ограничения реляционной модели, ограничив набор операций в ней семью основными и одной дополнительной.

Реляционная модель хранения данных построена на взаимоотношении составляющих ее частей. В простейшем случае она представляет собой двухмерный массив или двухмерную таблицу, а при создании сложных информационных моделей составляет совокупность взаимосвязанных таблиц.

Реляционная модель базы данных имеет следующие свойства:

1) каждый элемент таблицы – один элемент данных;

2) все столбцы в таблице являются однородными, ᴛ.ᴇ. имеют один тип (числа, текст, дата)

3) каждый столбец (поле) имеет уникальное имя;

4) одинаковые строки в столбце отсутствуют;

5) порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы – атрибутам отношений, доменам, полям.

В случае если реляционная модель данных состоит из нескольких таблиц, то они связываются между собой ключами.

Ключ – поле, ĸᴏᴛᴏᴩᴏᴇ однозначно определяет соответствующую запись (ключевое поле).

*Пример 3.* Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (рис.9).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ личного дела** | **Фамилия** | **Имя** | **Отчество** | **Дата рождения** | **Группа** |
| 25689 | Иванов | Иван | Иванович | 01.01.80 | 111 |
| 25231 | Петрова | Мария | Васильевна | 08.04.75 | 112 |
| 25549 | Сидоров | Василий | Петрович | 14.02.77 | 111 |

Рисунок 9 – Пример реляционной таблицы

Отношения представлены в виде *таблиц,* строки которых соответствуют кортежам или *записям,* а столбцы – атрибутам отношений, доменам, *полям*.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется *простым ключом* (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет *составной ключ.* В примере, показанном на рис.10, ключевые полем таблицы является «№ личного дела».

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы *внешний ключ –* ключ второй таблицы.

*Пример 4.* На рис.10 показан пример реляционной модели, построенной на основе отношений: СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ.

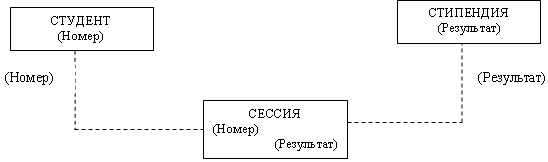
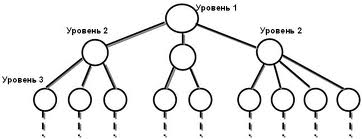


Рисунок 10 – Пример реляционной модели

***Иерархические БД***

В иерархической базе данных все элементы располагаются в последовательности от высшего к низшему и образуют перевернутое дерево (граф). Иерархическая БД имеет древовидную многоуровневую структуру.

.



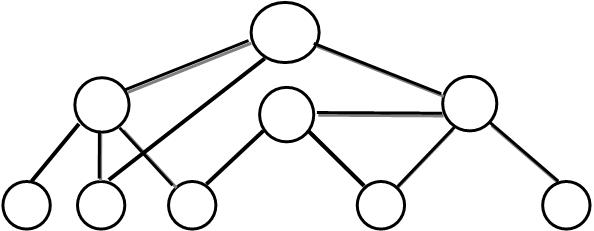
К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. *Узел –* это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем, и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи.

***Сетевые БД***

В сетевой базе данных к вертикальным иерархическим связям добавляются горизонтальные. Поэтому сетевая БД отличается большей гибкостью при поиске необходимых данных, представляет собой граф со свободным характером связей между узлами.

Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно - исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС. Графическое изображение описанной в примере сетевой структуры, состоящей только из двух типов записей. Единственное отношение представляет собой сложную связь между записями в обоих направлениях.



***Распределенные БД***

Распределенные - клиент – серверная технология.

**4. Система управления базами данных (СУБД).**

Система управления базами данных (СУБД) - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Состав СУБД

Обычно современная СУБД содержит следующие компоненты:

- ядро, которое отвечает за управление данными во внешней и оперативной памяти и [журнализацию](https://ru.wikipedia.org/wiki/%D0%96%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9),

- процессор языка базы данных, обеспечивающий [оптимизацию запросов](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2_%D0%A1%D0%A3%D0%91%D0%94) на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода,

- подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД

- а также сервисные программы (внешние [утилиты](https://ru.wikipedia.org/wiki/%D0%A3%D1%82%D0%B8%D0%BB%D0%B8%D1%82%D0%B0)), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы

Функции СУБД:

управление данными во внешней памяти (на дисках);

управление данными в оперативной памяти с использованием диского кэша;

журнализация изменений,резервное копирование и восстановление базыданных после сбоев;

поддержка языков БД (язык определения данных, язык манипулирования данных).

Объекты СУБД: таблицы, формы, запросы, отчёты, макросы и модули.

Таблицы - служат источником данных для запросов фирм, объектов.

В реляционных БД имеют двухмерную структуру.

Состоят из полей и записей.

Форма - это окно диалога или электронный документ, позволяющий осуществлять ввод данных и при задании определённых условий √ контролировать этот ввод.

Запросы - это средства отбора данных из одной или нескольких таблиц при помощи определённых пользователем условий. Запросы позволяют отбирать, добавлять, удалять данные. Запросы √ это тоже таблица, созданная только в памяти компьютера. В запросах можно создавать вычисляемые поля, по которым можно сортировать и отбирать данные.

Отчёт - это средство создания сложных многостраничных документов и вывода их на печать.

Макросы - содержат описания действий, которые должны быть выполнены, при совершение некоторых событий. Макросы создаются пользователем, (это программирование) на языке программирования (Visual Bai-sic).

Модули -это отдельные программы на языке Visual Bai-sic.

1. **Типы данных и их представление**

Человек воспринимает данные в виде образов, которые отображаются в его сознании. Компьютер воспринимает данные в виде двоичных кодов, поэтому для хранения и отображения данных в компьютере требуется определённое место памяти, а также инструкция по преобразованию двоичных кодов. Понятно, что для хранения простого числа потребуется значительно меньше места, чем для хранения фотографии, об этом следует постоянно помнить при создании базы данных, т.к. скорость обработки данных во многом зависит от вида их представления. В компьютере данные разделены на несколько типов, тип данных задаётся пользователем. Поэтому, при задании свойств полю таблицы, задают тип данных, которые будут представлены в столбце таблицы под общим заголовком (наименование поля). В Access 2010 существует 11 основных типов данных, а для некоторых из них введены подтипы данных, перечень типов данных приведён ниже.

*Текстовый.* Этот тип данных предназначен для обработки любых алфавитно-цифровых символов (включая числа), одиночных или собранных в строку. Максимальная длина строки не должна превышать 255 символов. Поэтому, при использовании данных типа текстовый целесообразно для каждого поля задавать его размер в символах (для экономии общего объёма памяти). По умолчанию всем полям таблицы в Access 2010 задаётся именно этот тип данных.

*Поле МЕМО.* Это текстовое поле очень большой длины. Обычно используется для хранения абзацев текста, резюме, кратких описаний и т.п. При использовании этого типа нет необходимости заботиться об указании длины записи, которая будет храниться в памяти машины. Для поля с типом данных МЕМО выделяется ровно столько памяти, сколько требуется для конкретной записи.

*Числовой.* СУБД Access позволяет выполнять над полями такого типа вычисления. Чтобы указать конкретный способ хранения данных в полях числового типа, предусмотрено несколько подтипов:

1. Байт – занимает 1 байт.

2. Целое – занимает 2 байта.

3. Длинное целое – занимает 4 байта.

4. Одинарное с плавающей точкой – занимает 4 байта.

5. Двойное с плавающей точкой – занимает 8 байт.

6. Действительное – занимает 12 байт.

*Денежный.* В таком поле хранятся данные, представленные согласно правилам бухгалтерии (обозначение валюты, разделение числа на разряды), хотя, с такими данными выполняются все арифметические и логические операции, как с типом данных – числовой. В памяти отображение записи такого типа занимает 8 байт.

*Дата и время.* Это специальный тип данных, в котором хранятся значения даты или времени, или и того и другого. Для разнообразного отображения даты и времени существуют специальные форматы в виде шаблонов, что позволяет проводить вычисления над этими данными. В памяти машины такой тип данных занимает 8 байт.

*Счётчик.* Тип данных, который используется для идентификации записи в базе данных. Обычно счётчик используют в качестве ключевого поля в таблицах, поэтому он представляется в виде целого числа.

*Логический.* Такие поля содержат значения: Да\Нет, Истина\Ложь, Вкл.\Выкл. и т.п. Фактически, в этом поле 1 интерпретируется как Да, а 0 – как Нет.

*Гиперссылка.* Обеспечивает связь с Web-страницей или каким-либо файлом. При выборе этого типа данных Access автоматически запускает приложение, в котором можно отобразить этот файл, например Web-браузер для отображения Web-страницы или программный модуль с расширением .exe. Поле объекта OLE (Object Linking and Embedding – связывание и внедрение объекта). В поле могут быть размещены рисунки, электронные таблицы, видеофильмы в двоичном формате. Такой тип данных сохраняет данные объёмом до 1 Гбайт.

*Вложение.* Тип данных, который появился в Access 2007|2010 для хранения непосредственно в базе данных файлов больших размеров (рисунков, приложений MS Office и других видов) в сжатом виде размером до 2 Гбайт. Мастер подстановок. Это свойство поля, а не настоящий тип данных используется в Access 2010 в качестве средства для более эффективного и корректного ввода данных. При выборе этого типа данных запускается мастер подстановок, который создаёт поле в том формате и того типа, которое будет позаимствовано из другой таблицы или списка, обычно этот тип данных занимает 4 байта.

В Access 2010 предусмотрены варианты задания форматов полей таблицы непосредственно при её создании. Когда открывается режим создания таблицы, то можно сразу задавать необходимый формат полю. На рисунке 5 показано, что при нажатии на стрелку в строке с наименованием полей таблицы, будет раскрыт список с видами форматов, которые можно установить (по умолчанию формат поля таблицы задаётся, как – Текст).

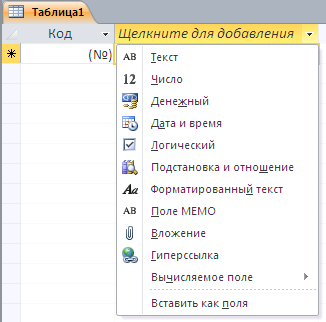


Рис. 5. Список с наименованием форматов для поля таблицы

**Контрольные вопросы**

1. Что представляют собой базы данных?
2. Приведите пример известных баз данных?
3. Для чего применяются базы данных?
4. Какими возможностями обладают базы данных?
5. Какие бывают модели баз данных?
6. Для чего в приложениях MS Access используют различные типы данных?
7. Какие типы данных содержат символьную и текстовую информацию?
8. Чем отличаются типы данных, которые могут содержать числа от типов данных, которые могут содержать символьные данные?
9. Как воспринимает система логический тип данных?
10. Какой из типов данных наиболее приемлем для хранения изображений?

**Рекомендуемая литература**

1. Зотова С.И. Практикум по Access - М., Финансы и статистика, 2007
2. Практикум по экономической информатике. Учебное пособие под редакцией проф. Шуремова Е.Л. –М:,2014
3. Быковская Т.И. Экономическая информатика и информационные технологии: лабораторный практикум., Фолиант, Астана, 2011