



Лекция 3

План лекции

Запись и хранение данных.

Массивы, списки, очереди, стеки,
файловый ввод-вывод.

Основные понятия о базах данных.



Запись и хранение данных

Варианты записи и формы хранения данных зависят от их характеристик (типа, объема данных, периодичности их поступления, скорости передачи данных, быстродействия измерительной системы, периода использования данных).



Запись и хранение данных

В зависимости от решаемых задач различают сохранение данных:

- **кратковременное** (для краткосрочных целей – используется хранение в форме различных структур данных в оперативной памяти компьютера);
- **долговременное** (предполагается в последующем многократный доступ к данным – используется хранение данных на жестком диске и других носителях для сохранения данных на длительный период).



Массивы

Массивы используются при хранении данных, объем которых заранее известен или ограничен верхней границей

Обеспечивают **произвольный доступ** к элементам по индексу.

Элементы массивов также могут быть массивами (т.наз. многомерные массивы)



Массивы – статические и динамические

```
Dim A(10) As Double    ` статический
```

```
Sub Main()
```

```
.....
```

```
Dim MyArray() As Integer ` динамический
```

```
Sub Main()
```

```
.....
```

```
Redim MyArray(5)
```

```
For I = 1 To 5
```

```
    MyArray(I) = I
```

```
Next I
```

```
Redim MyArray(10)
```

```
.....
```



Списки

Списки используются для хранения данных, **объем которых не известен (может динамически изменяться)**.

Память для хранения данных при этом динамически выделяется (или возвращается).

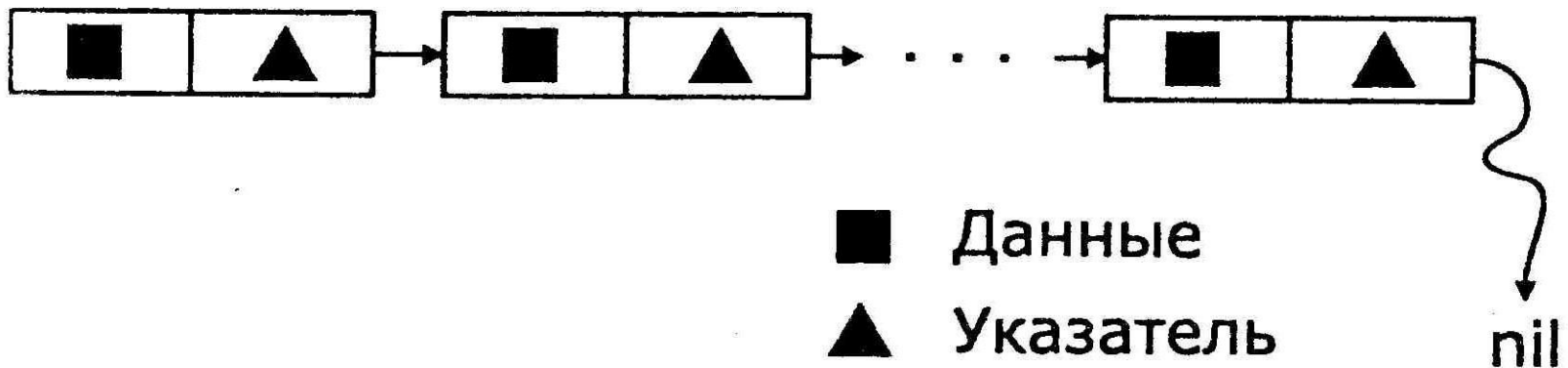
Описанная структура обеспечивает таким образом, **последовательный доступ** к элементам.

При передаче данных во вторичную память в этом случае требуется **буфер** для передачи части потока

Линейные списки

Линейный список (очередь) – последовательность однотипных данных, в котором каждое звено ссылается на следующее.

Последнее звено ни на что не ссылается (NIL).



Структура линейного списка



Линейные списки

Указатели в языках Pascal, Delphi:

$P^{\wedge} := 1;$ // переменной, на которую указывает P, присваивается 1

$A := P^{\wedge};$ // Переменной A присваивается содержимое переменной, на
// которую указывает P

Построение линейного списка с помощью указателей

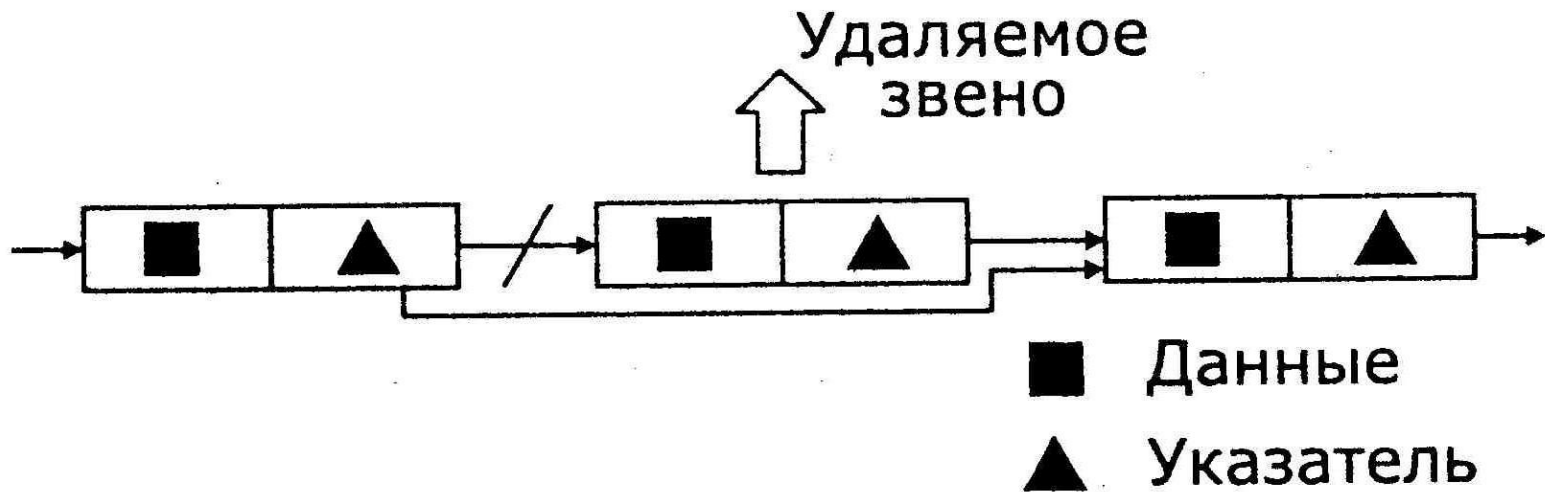
$S := \text{nil};$ // список пустой, указатель на головное звено пустой

$\text{New}(SS);$ // отвести в памяти область для нового звена

$SS^{\wedge} := 123;$ // присвоить звену значение

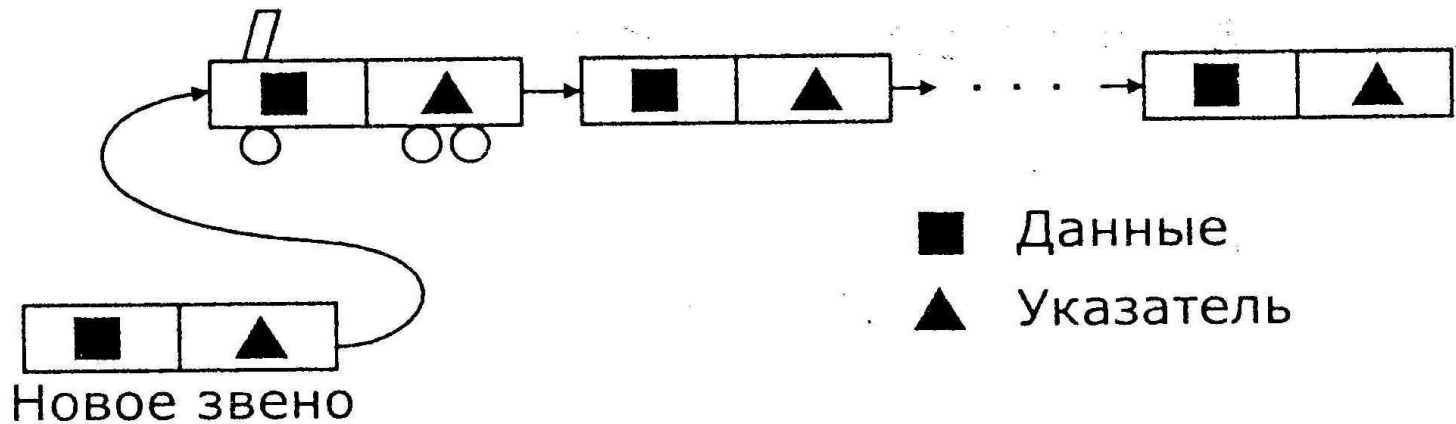
$SS^{\wedge}.\text{Next} := S;$ // сформировать ссылочную часть нового звена

Линейные списки



Удаление промежуточного звена из
линейного списка

Линейные списки

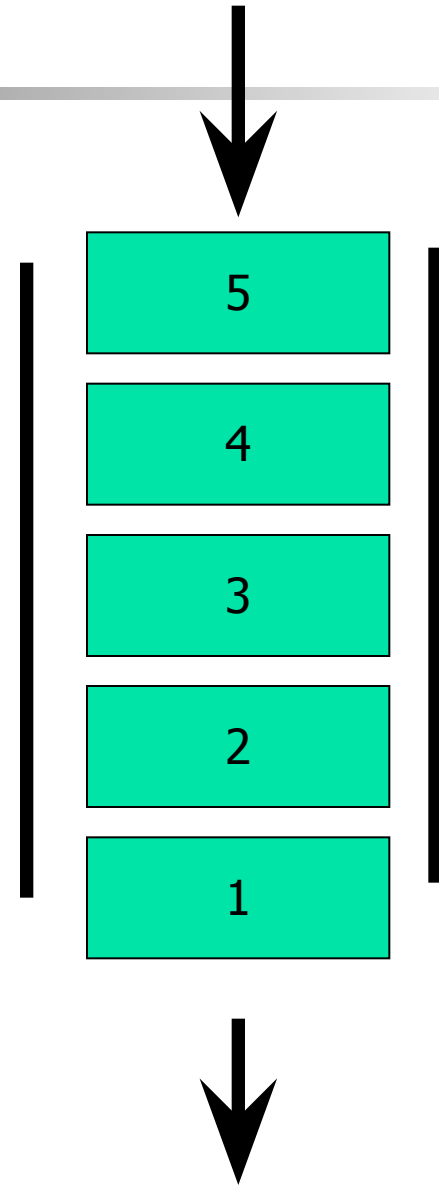


Добавление нового звена в линейный список

Очередь

Очередь (FIFO – «First In – First out») –

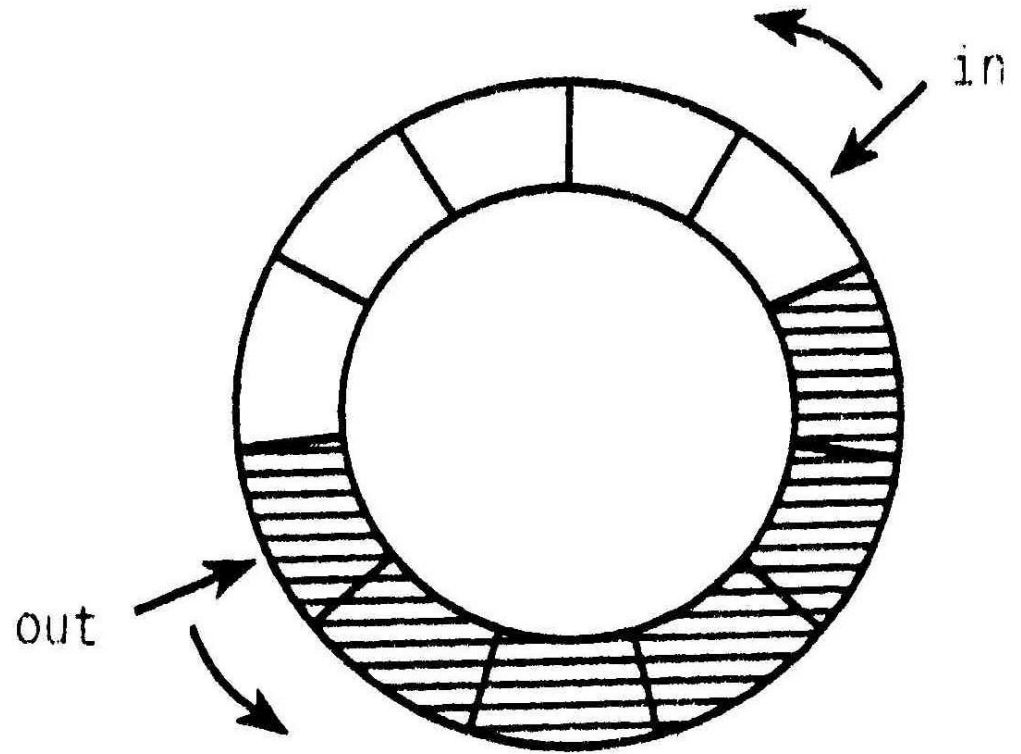
используется для
записи подзадач,
выполнения
арифметических
действий



Очередь

Круговой буфер,
очередь (FIFO –
«First In – First
out») –

используется для
отображения
данных Real Time
(бегущий экран
осциллографа).





Очередь

```
NBuf:=1024; // размер  
           // буфера
```

// размещение в буфере:

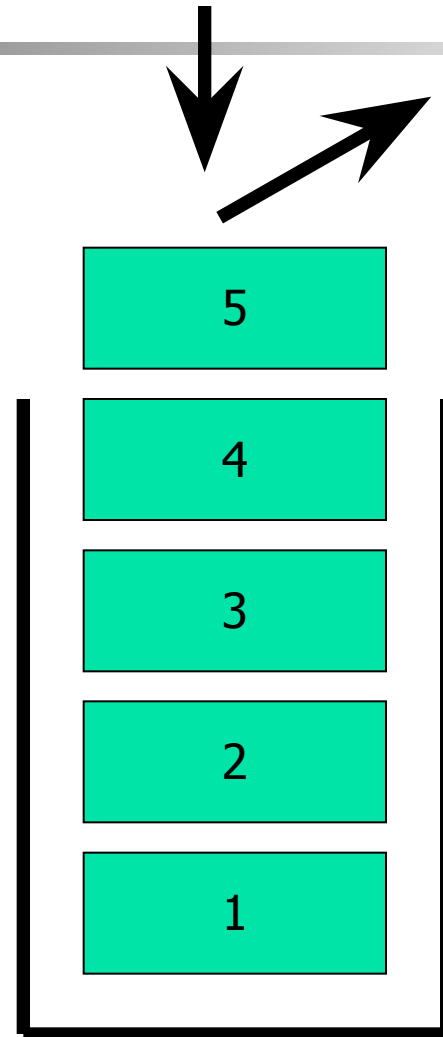
```
n:=n+1;  
If n<NBuf then  
begin  
    in:=(in + 1) MOD N  
    buf [in]:=x;  
end;
```

// извлечение из буфера:

```
n:=n-1;  
If n<NBuf then  
begin  
    out:=(out + 1) MOD N  
    x:=buf [out];  
end;
```

Стеки

Стек (LIFO – «Last In – First out») –
используется для
записи подзадач,
выполнения
арифметических
действий





Файловый ввод-вывод

Для данных последовательного доступа (диски, ленты и т.п.) используется запись в **файл последовательного доступа**.

Для произвольного доступа к данным – **файл произвольного доступа**, содержащий адресацию.

Последовательный доступ позволяют использовать все внешние устройства.

Произвольный доступ - только магнитные и оптические диски, флэш-накопители.



Файловый ввод-вывод

Данные хранятся в файлах, как правило, в **бинарной** (набор символов «0» и «1», для представления которых используется 1 бит) или

текстовой форме (кодировка **ASCII** - **American Standard Code for Information Interchange**, для представления каждого из 256 символов которой требуется 1 байт = 8 бит).



Файл последовательного доступа

Это в основном **текстовые** файлы, содержащие коды символов, **признаки перевода строки и конца файла**. Записи – строки **переменной** длины, отделенные символом перевода строки. Обычно создаются **для обработки и хранения текстовой информации (но не числовой)**.
Читаются от начала к концу, поэтому **невозможно одновременно считывать из них данные, и записывать их**. Чтобы изменить одну запись файла, его нужно **записать заново**.



Файл произвольного доступа

Используются, если **требуется частый доступ к данным**. Как и в файлах с последовательным доступом, текстовые данные хранятся в них в виде символов. Однако числа хранятся в естественном формате (Integer, Double, Single и т. д.). **Позволяют в любой момент обработать любую запись в файле по ее индексу**. Такие файлы можно открывать для **одновременного чтения и записи**.



Бинарные (двоичные) файлы

Двоичные файлы подобны файлам с последовательным доступом, но **длина записи у этих файлов равна одному байту**



Базы данных

База данных (БД) – это хранилище данных о некоторой предметной области, организованное в виде специальной структуры.

Система управления базой данных (СУБД) – это программное обеспечение для работы с БД.

Функции:

- поиск информации в БД

- выполнение несложных расчетов

- вывод отчетов на печать

- редактирование БД



Типы информационных систем

локальные ИС

БД и СУБД находятся на одном компьютере.

файл-серверные

БД находится на сервере сети (файловом сервере), а СУБД - на компьютере пользователя.

клиент-серверные

БД и основная СУБД находятся на сервере, СУБД на рабочей станции посылает запрос и выводит на экран результат.



Типы баз данных

Реляционные

Табличные
(данные в виде одной таблицы)

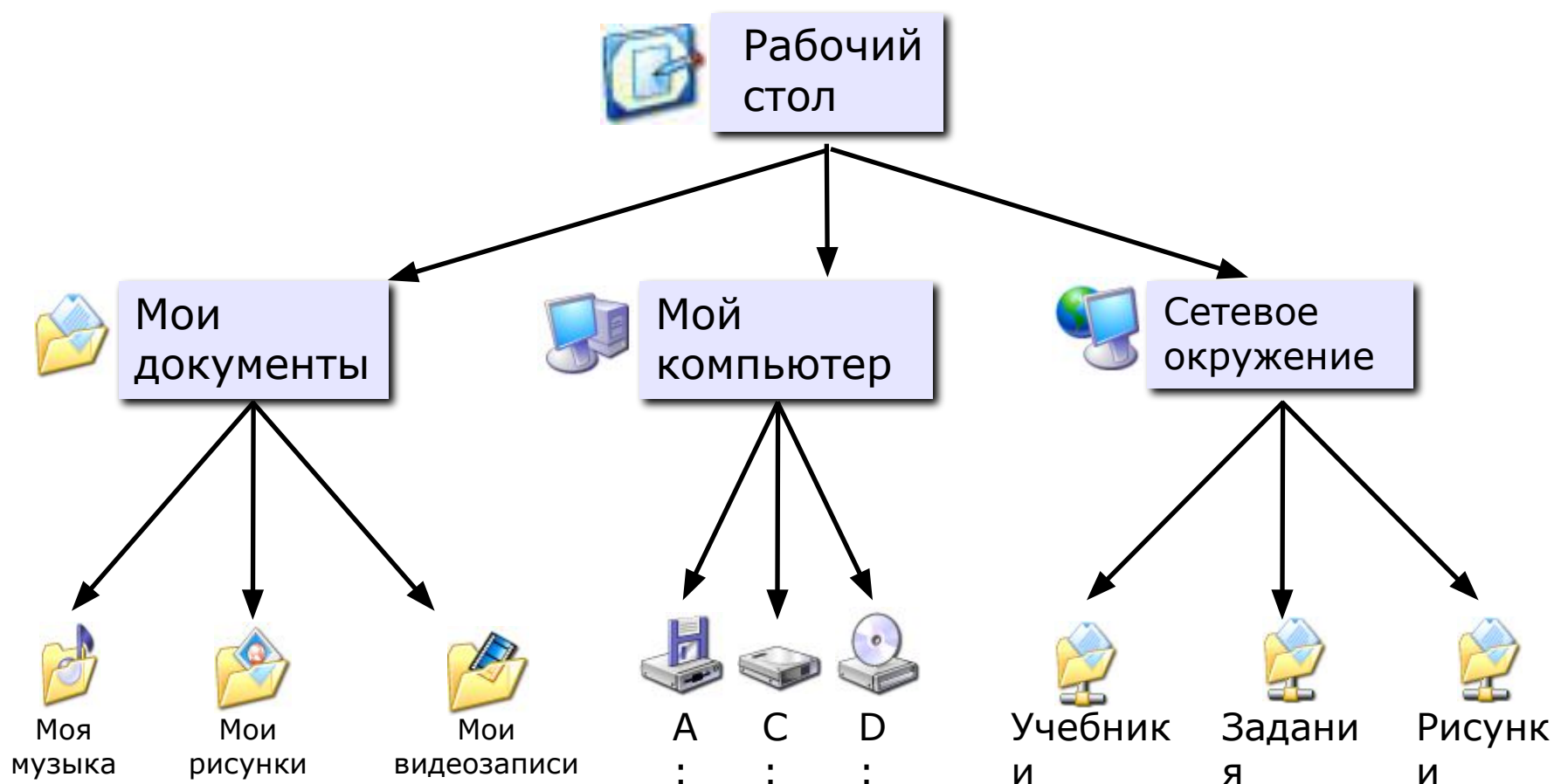
Многотабличные
(множество взаимосвязанных таблиц)

Нереляционные

Иерархические
(в виде многоуровневой структуры)

Сетевые
(набор узлов, в котором
каждый может быть связан с
каждым)

Иерархические базы данных



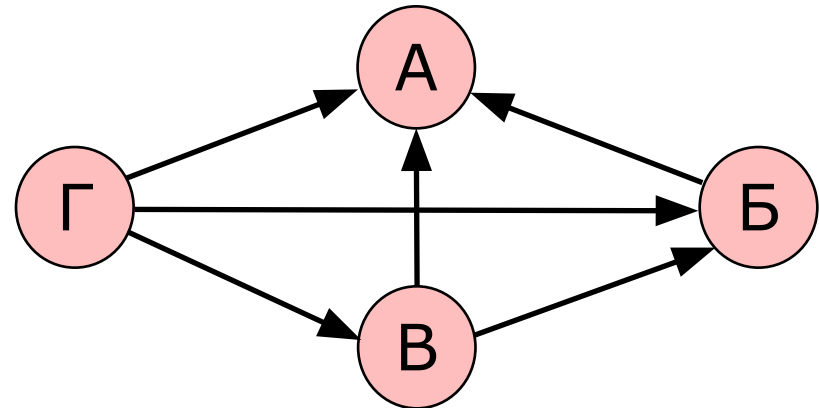
Сетевые базы данных

Сетевая БД – это набор узлов, в которых каждый может быть связан с каждым (схема дорог).

Сетевая БД наилучшим образом отражает структуру некоторых задач (сетевое планирование в экономике)

Недостатки Сетевых БД :

- сложно хранить информацию о всех связях;
- запутанность структуры.





Реляционные базы данных

Поле БД – это столбец таблицы, содержащий значения определенного свойства

Запись БД – это строка таблицы, содержащая набор значений свойств, размещенный в полях БД

Ключевое поле (первичный ключ) - это поле (или комбинация полей), которое однозначно определяет запись.

В таблице не может быть двух записей с одинаковым значением ключа.

Реляционные базы данных

Microsoft Access

Файл Правка Вид Вставка Формат Записи Сервис Окно Справка



DB-2013 : база данных (формат Access 2000)

Сотрудники : таблица

	ID	Фамилия	Имя	Отчество	Факультет	Кафедра	Должность	Ставок	Нагрузка_общ	С
▶	184	Абабурко	Виктор	Николаевич	ЭТФ	ЭП	Ст.преп.	1	930	
	304	Александрова	Светлана	Александровна	ЭФ	КД	Ст.преп.	1	925	
	221	Александрова	Светлана	Александровна	ЭФ	КД	Ст.преп.	1	955	
	564	Александрова	Светлана	Александровна	ЭФ	КД	Ст.преп.	1	1370	
	3	Алексо	Ольга	Анатолевна	ИЭФ		методист	1	0	
	340	Алексютина	Галина	Яковлевна		ГД	Ст.преп.	1	932	
	80	Алексютина	Галина	Яковлевна		ГД	Ст.преп.	1	1379	
	487	Алексютина	Галина	Яковлевна		ГД	Ст.преп.	1,25	1265	
	622	Алехнович	Светлана	Владимировна	СФ	СКЗС	Асс.	0	433	
	474	Алехнович	Людмила	Адамовна	СФ	НачГеог	Преп.	0,5	490	
	137	Алехнович	Людмила	Адамовна	СФ	НачГеог	Преп.	0,5	498	
	488	Артемчик	Виталий	Викторович		ГД	Ст. пр.	1,25	1010	
	81	Артемчик	Виталий	Викторович		ГД	Ст.преп.	1,25	1160	
	341	Артемчик	Виталий	Викторович		ГД	Ст.преп.	1	1010	
	8	Аталь	Евгения	Викторовна	АМФ	БРИЯ	Ст. преп.	1	1097	
	364	Афанасенко	Татьяна	Васильевна		ФиС	Ст.преп.	1	990	
	593	Афанасенко	Татьяна	Васильевна	ЭФ	ФБУ	Ст.преп.	1	990	



Табличные базы данных

- 1.Количество полей** определяется разработчиком и не может изменяться пользователем.
- 2.Любое поле должно иметь **уникальное имя.**



Табличные базы данных

3. Поля могут иметь различный **тип**:

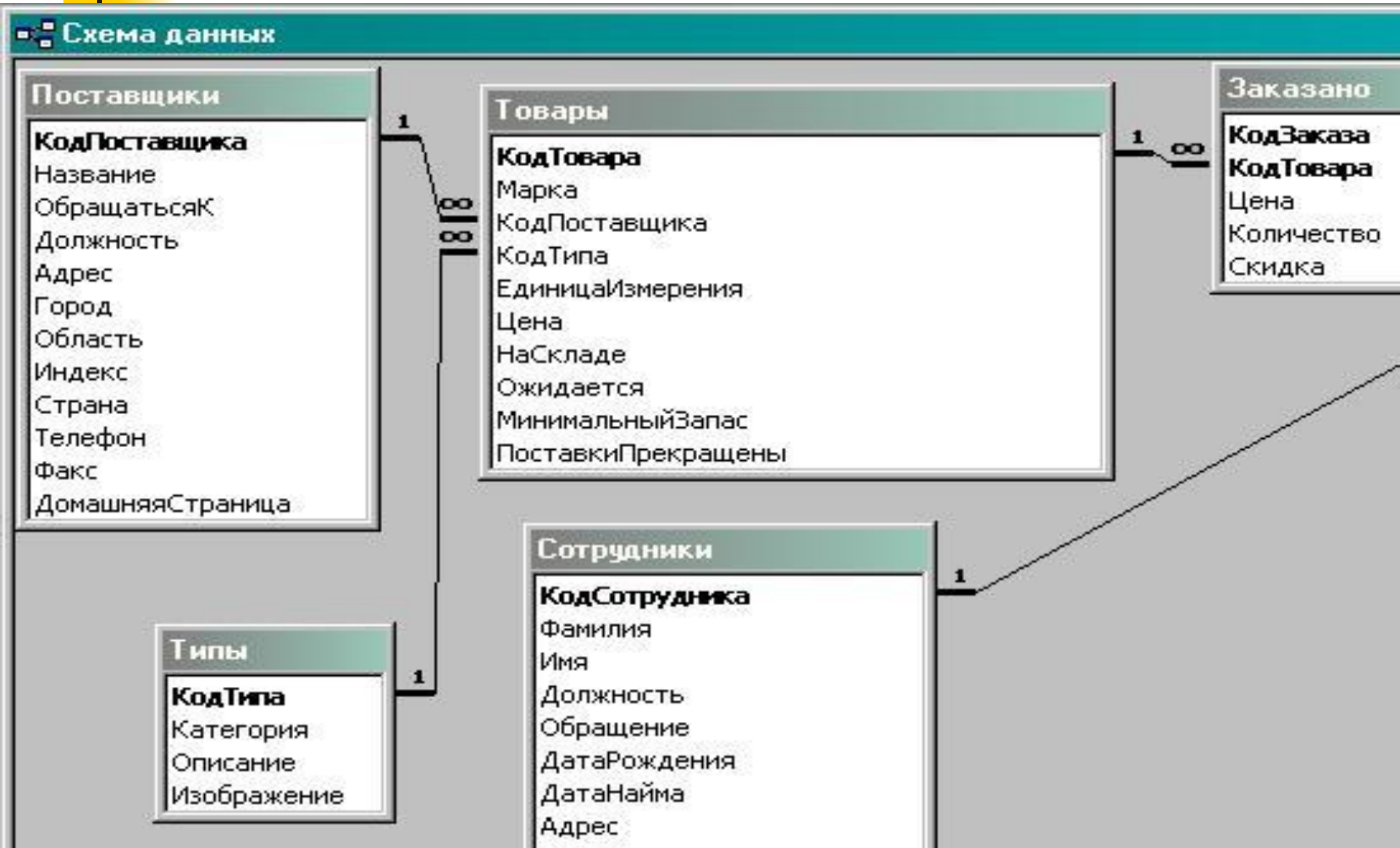
- строка символов (длиной до 255 символов)
- вещественное число (с дробной частью)
- целое число
- денежная сумма
- дата, время, дата и время
- логическое поле (истина или ложь, да или нет)
- многострочный текст (MEMO)
- рисунок, звук, другой объект (объект OLE)



Табличные базы данных

1. Поля могут быть **обязательными для заполнения или нет.**
2. Таблица может содержать **сколько угодно записей** (это количество ограничено только объемом диска); записи можно добавлять, удалять, редактировать, сортировать, искать.

Реляционные базы данных





Язык SQL и пример запроса

Пример для вывода в MS Access информации о поступлении на склад фрезы 12-го июня 2013 года:

```
SELECT dt, product_id, title, amount, price  
FROM m_income INNER JOIN m_product  
ON m_income.product_id=m_product.id  
WHERE title='Фреза' And dt=#6/12/2013#;
```

Искомый текст заключен в одинарные кавычки (').