Лекция 2
Системы управления базами данных

1. Особенности реляционных баз данных.
2. Постреляционная модель представления данных.
3. Объектно-ориентированные технологии в базах данных.
4. Объектно-реляционные среды и методы.

Основные определения

- База данных

  Это большой массив информации (совокупность сведений) о конкретных объектах реального мира в какой-либо предметной области

- СУБД (Система управления базами данных

  Это совокупность программных средств, обеспечивающая возможность создания базы данных, доступа к данным и управление базой данных.

Функции СУБД

- управление данными во внешней памяти
- управление данными в оперативной памяти
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев
- поддержка языков БД (язык определения данных, язык манипулирования данными).

Типы структур (моделей) БД

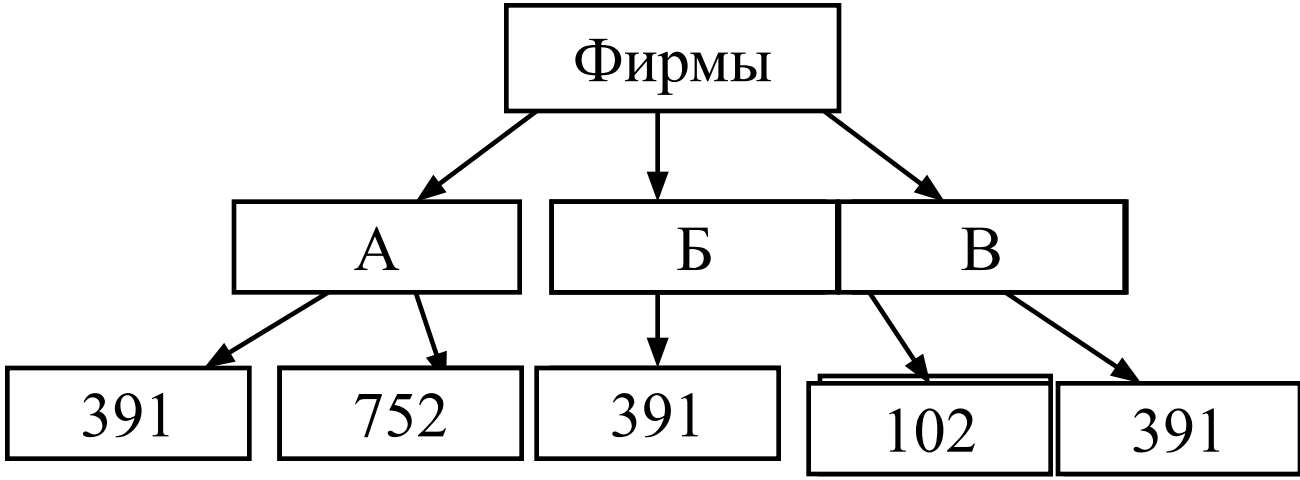
- Двухмерная или табличная (или реляционная)
- Иерархическая
- Сетевая

Типы структур БД

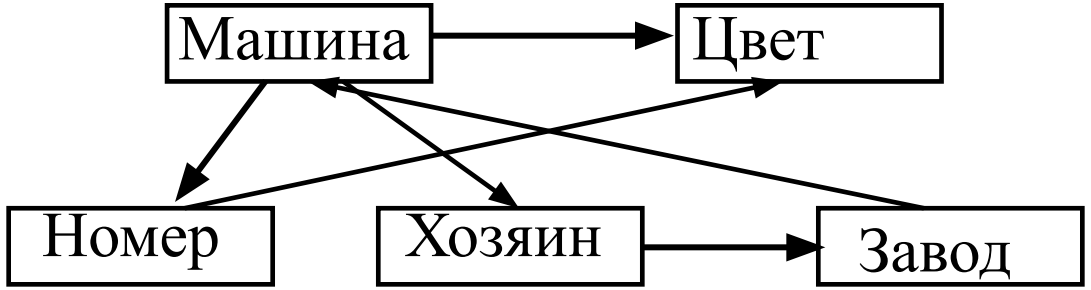
Реляционная

Телефон	ФИО	Адрес
25-25-25	Иванов	Кемерово

Иерархическая



Сетевая



Типы структур БД

Для **иерархических структур** характерна подчиненность объектов нижнего уровня объектам верхнего уровня. В дереве, между верхними и нижними объектами, задано отношение **«один ко многим».** Исходные элементы порождают подчиненные.

Сети имеют много уровней взаимосвязанных объектов, между которыми задано отношение **«многие ко многим».** Сетевая организация обладает большей гибкостью и облегчает процесс поиска требуемых данных.

Реляционные базы данных

Реляционные базы данных получили наибольшее распространение, т.к. они обладают преимуществом - наглядность и понятность для пользователя табличной структуры.

К реляционной структуре можно свести любой тип структуры данных (деревья и сети).

Название “**реляционная**” (от relational - отношение) связано с тем, что каждая запись в таблице содержит информацию, относящуюся только к одному конкретному объекту.

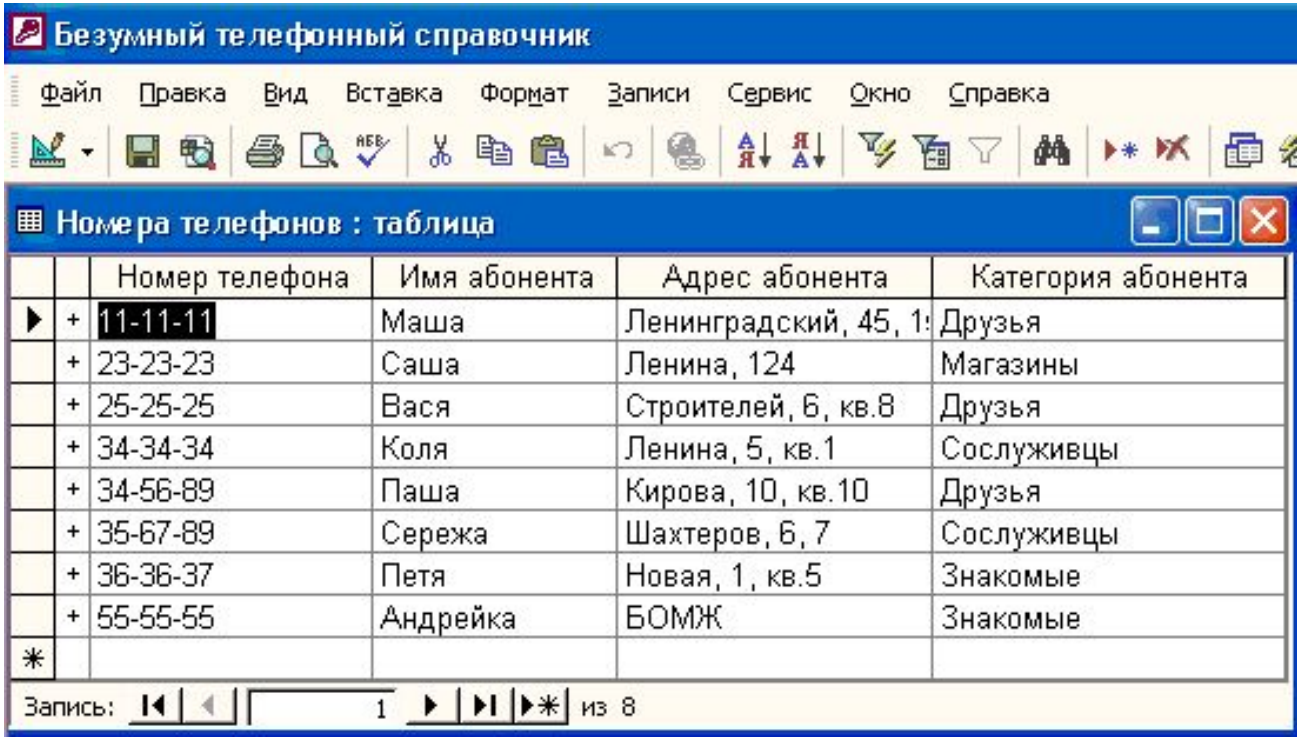
Чаще всего база данных строится на основе нескольких таблиц, связанных между собой.

Реляционные базы данных

- Все данные в реляционной БД представлены в виде таблиц. Каждая строка таблицы содержит информацию только об одном объекте и называется **записью**. Столбец таблицы содержит однотипную для всех записей информацию и называется **полем**. Для успешного функционирования базы данных важна правильная организация данных в ней. При определении структуры данных в базе выделяют следующие основные понятия.
- **Класс объектов** - совокупность объектов, обладающих одинаковым набором свойств. ***Например***, в базе данных о ВУЗе классами объектов могут быть студенты, преподаватели, предметы.
- **Свойство (атрибут)** - определенная часть информации о некотором объекте. Хранится в виде столбца (поля) таблицы. ***Например***, фамилия, имя, отчество - это свойства для объекта Студент.

Понятие **ЗАПИСИ** БД

Это 1-ая запись



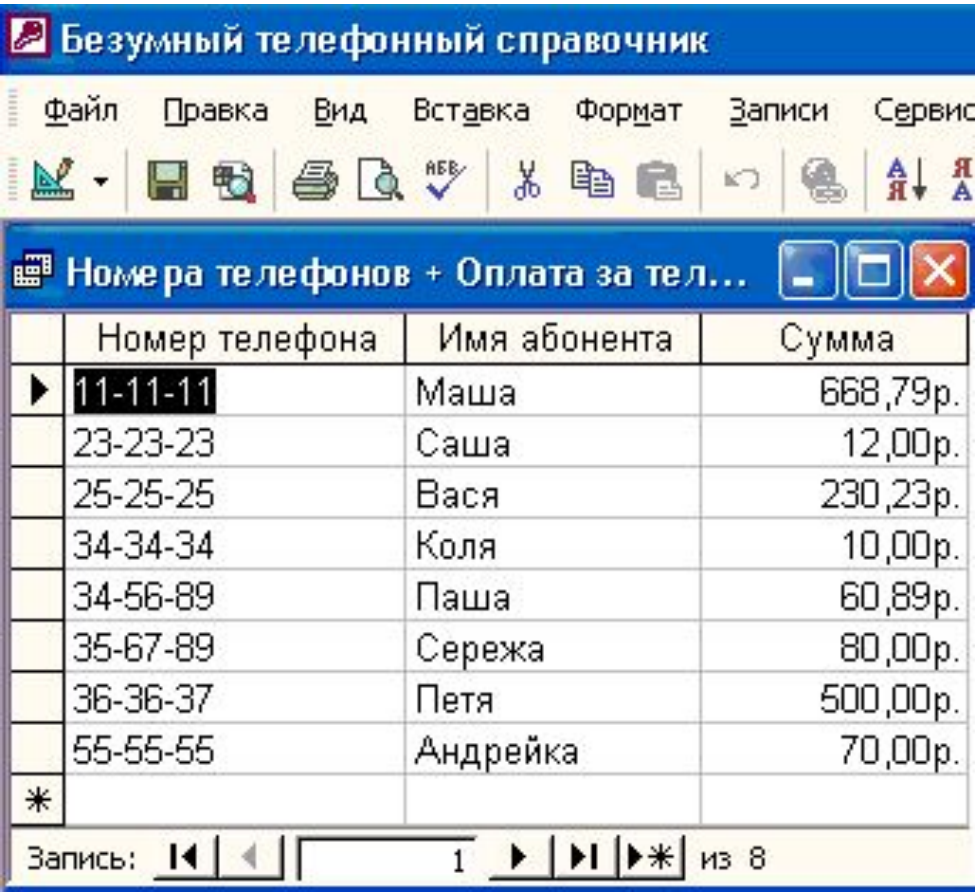
	Номер телефона	Имя абонента	Адрес абонента	Категория абонента
+	11-11-11	Маша	Ленинградский, 45, 1	Друзья
+	23-23-23	Саша	Ленина, 124	Магазины
+	25-25-25	Вася	Строителей, 6, кв.8	Друзья
+	34-34-34	Коля	Ленина, 5, кв.1	Сослуживцы
+	34-56-89	Паша	Кирова, 10, кв.10	Друзья
+	35-67-89	Сережа	Шахтеров, 6, 7	Сослуживцы
+	36-36-37	Петя	Новая, 1, кв.5	Знакомые
+	55-55-55	Андрейка	БОМЖ	Знакомые
*				

Запись: 1 из 8

Каждая запись должна иметь свой уникальный номер

Каждая *строка* таблицы БД содержит один блок данных и представляет собой **запись**.

Понятие **ПОЛЯ** БД



Колонки в таблице БД называются **ПОЛЯМИ**

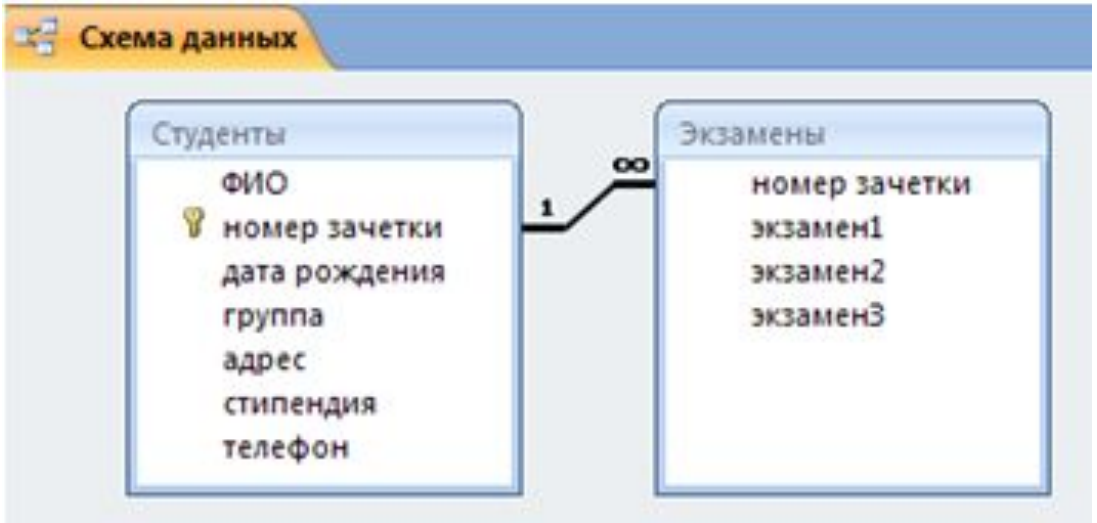
Любое поле имеет свое уникальное **имя**

В Access имена полей - до 256 символов.

Количество полей, имена, типы данных задаются при формировании **структуры таблицы**

Связи между таблицами

Связи между любыми двумя таблицами относятся к одному из трех типов: один-к-одному (1:1), один-ко-многим (1:М) и многие-ко-многим (М:М).
При установке связи типа **"один-к-одному"** (1:1) каждой записи в одной таблице соответствует не более одной записи в другой таблице.
Связь типа **"один-ко-многим"** (1:М) означает, что каждой записи в одной таблице соответствует несколько записей в связанной таблице. Этот наиболее распространенный тип связей. Для его реализации используются две таблицы. Одна из них представляет сторону "один", другая – сторону "много".
Связь типа **"много-ко-многим"** (М:М) используется, когда множеству записей в одной таблице соответствует множество записей в связанной таблице.




Тип определяет множество значений, которые может принимать данное поле в различных записях
Числовой
Значение поля может быть только числом
Символьный
Символьные последовательности (слова, тексты, коды и т.п.)
Дата / время
Календарные даты ДД/ММ/ГГ (ДД.ММ.ГГ)
Время суток ЧЧ:ММ (ЧЧ:ММ:СС)
Логический
да
нет
true
false
1
0

Типы данных

- ***Текстовый*** – одна строка текста до 255 символов
- ***Поле МЕМО*** – текст из нескольких строк с полосой прокрутки до 65535 символов
- ***Числовой*** – число любого типа (целое, вещественное и т. д.)
- ***Дата/время*** – поле, содержащее дату или время
- ***Денежный*** – поле, выраженное в денежных единицах (рубли, $ и т.д.)

Типы данных

- ***Счетчик*** – поле, которое вводится автоматически с вводом каждой записи, служит для нумерации записей
- ***Логический*** – содержит одно из значений True или False
- ***Поле объекта OLE*** – содержит рисунки, звуковые файлы, таблицы Excel и т.д.
- ***Гиперссылка*** – поле для хранения URL-адресов Web-страниц

Первичный ключ БД

Первичным ключом **в базе данных называют поле (или совокупность полей), значение которого не повторяется у разных записей.**

В БД «Детская библиотека» разные книги могут иметь одного автора, могут совпадать названия книг, год издания, полка. Но инвентарный номер у каждой книги свой (поле НОМЕР). Он-то и является первичным ключом для записей в этой базе данных.

Первичным ключом в БД «Погода» является поле ДЕНЬ, так как его значение не повторяется в разных записях.

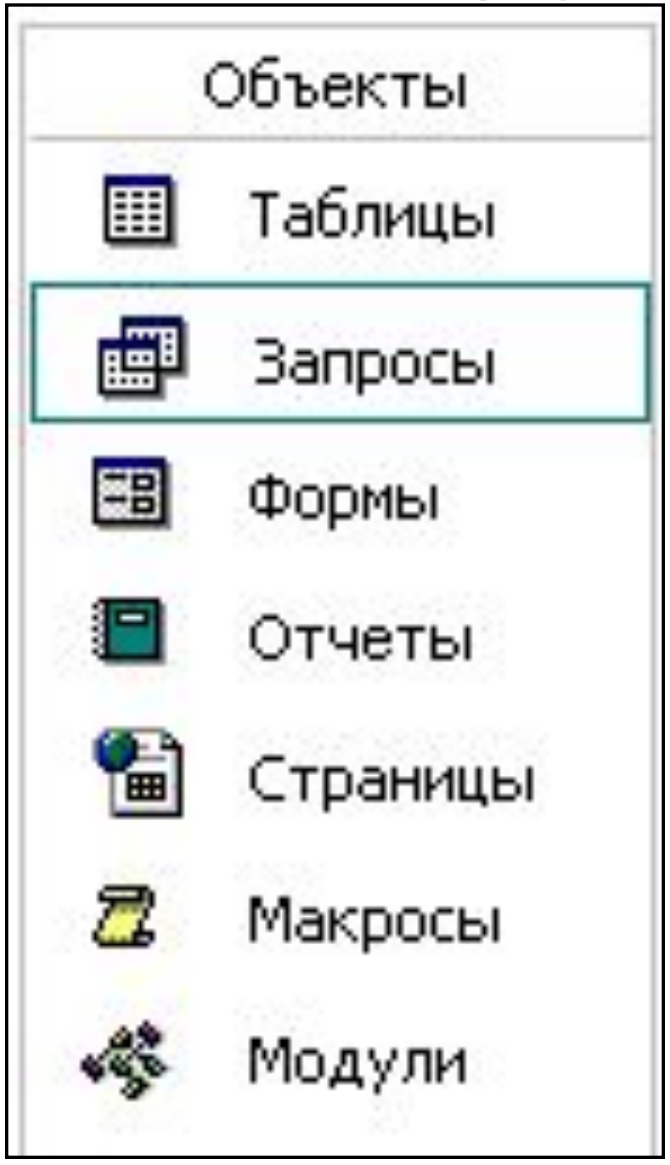
- Каждая таблица в реляционной базе данных должна иметь уникальный (первичный) ключ, однозначно определяющий каждую запись в таблице. Это позволяет быстро найти нужную запись, а также связать данные из разных Таблиц в запросах, формах и отчетах.

Имя поля	Тип данных
Код	Счетчик
№	Числовой
песня	Текстовый
слова	Текстовый
музыка	Текстовый
исполнитель	Текстовый
звание исполнителя	Текстовый

В качестве первичного ключа может быть задано поле с типом данных **Счетчик.** В этом случае при добавлении каждой новой записи в таблицу в это поле автоматически вводятся уникальные целые последовательно возрастающие (на 1) или случайные числа. Указание такого поля является наиболее простым способом создания первичного ключа. Значение этого поля нельзя изменить или удалить.

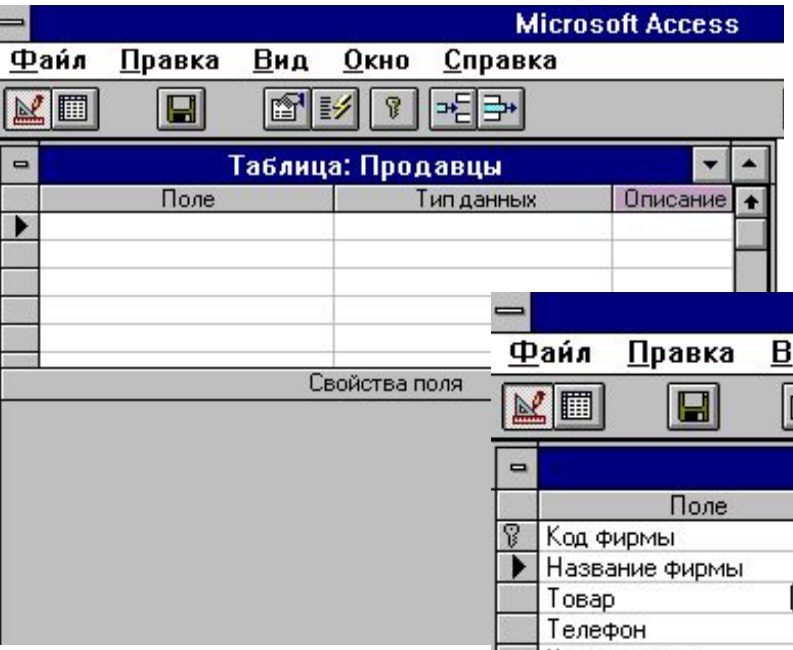
*

Объекты ACCESS

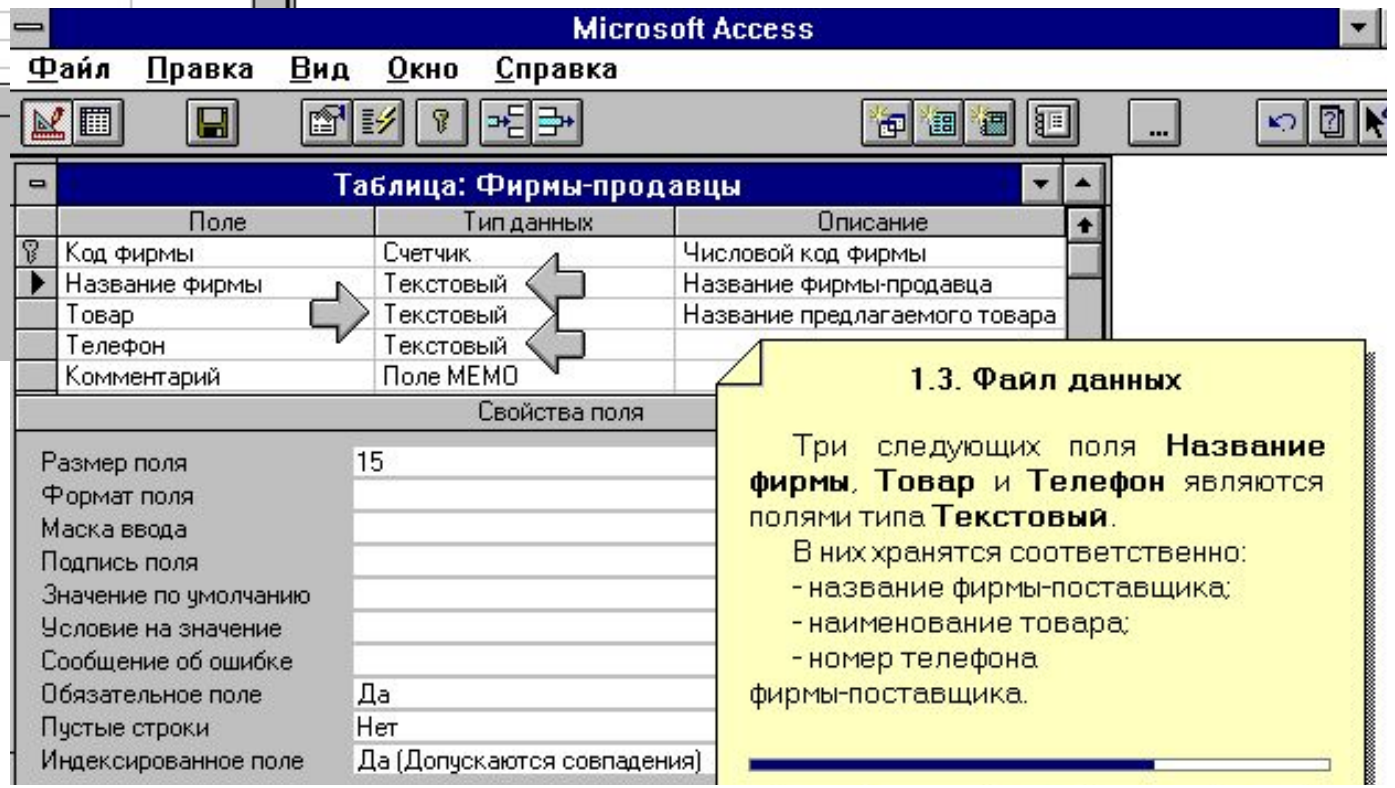


Access представляет большой выбор способов хранения данных.
Компоненты, которые используются для хранения и представления данных называются **объектами**.

Таблицы в Access



Создание БД начинается с описания **структур** всех таблиц, которые образуют БД



Формы

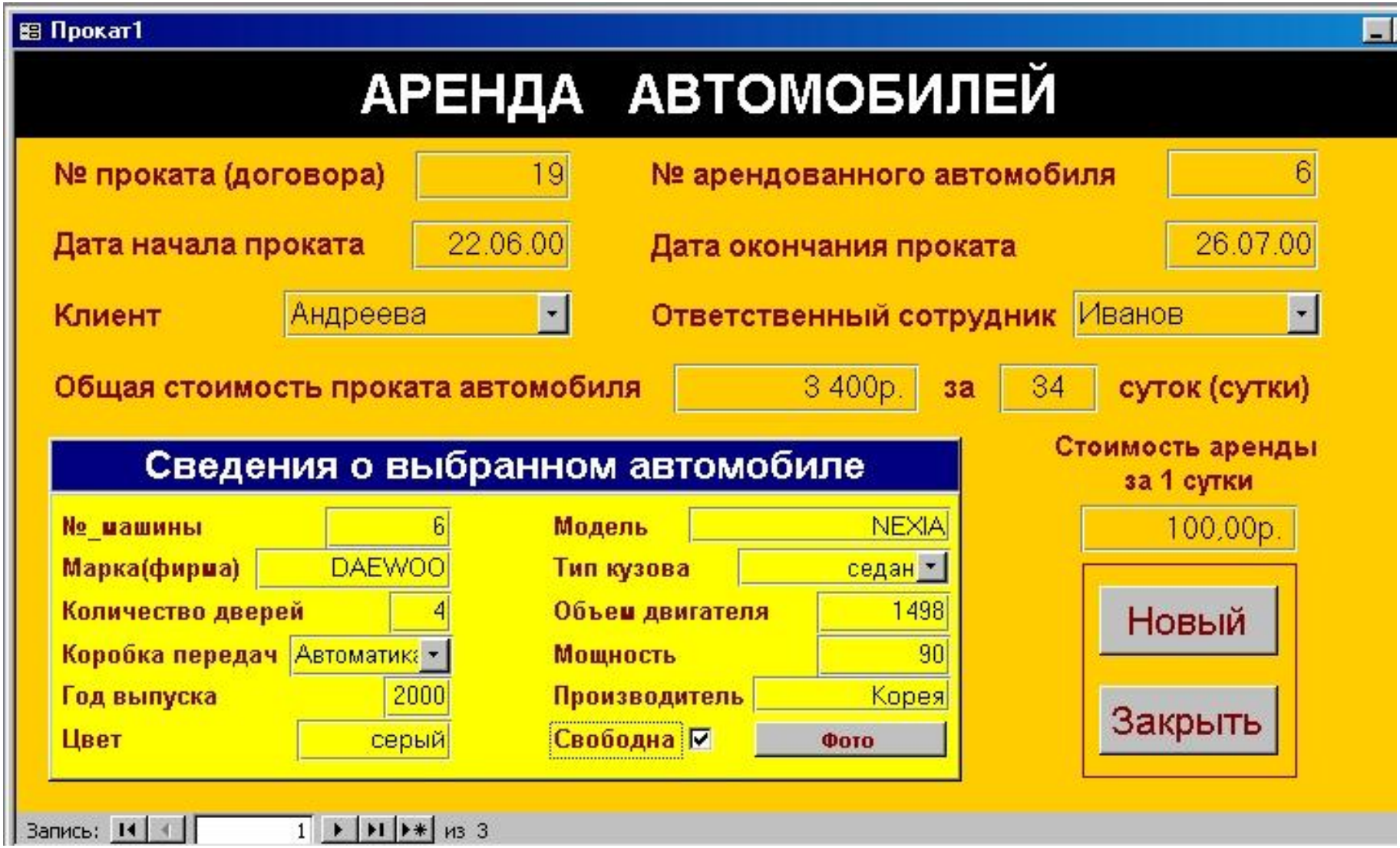
Позволяют удобно вводить и отображать данные, хранящиеся в отдельных записях.

Формы позволяют видеть столько данных из таблицы, сколько вы пожелаете.

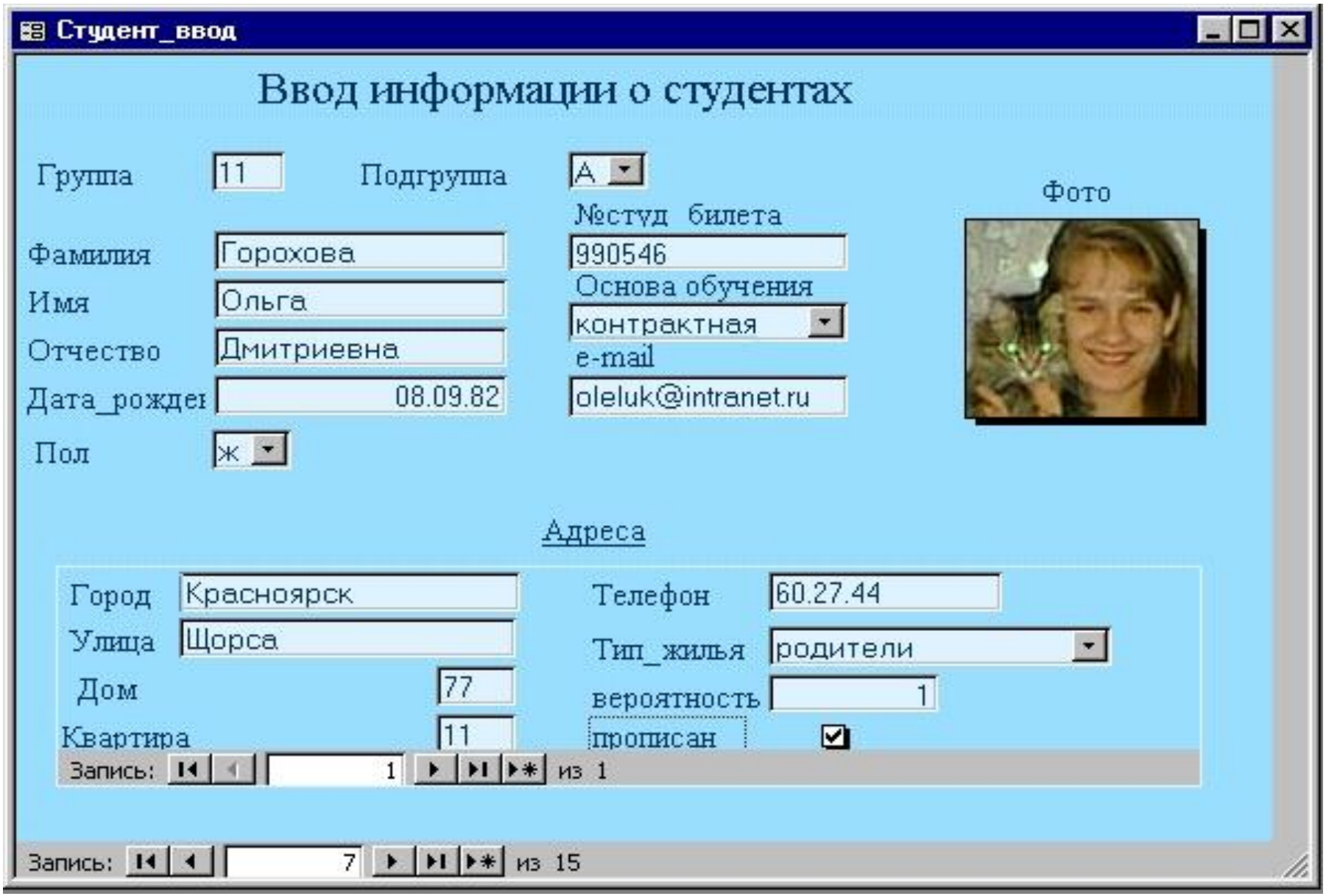
Формат представления выбирается по собственному желанию.

Если вы редактируете данные в таблице, то Access обновляет соответствующую информацию в форме.

Формы


Прокат1
АРЕНДА АВТОМОБИЛЕЙ
№ проката (договора)
19
№ арендованного автомобиля
6
Дата начала проката
22.06.00
Дата окончания проката
26.07.00
Клиент
Андреева
Ответственный сотрудник
Иванов
Общая стоимость проката автомобиля
3 400р.
за
34
суток (сутки)
Сведения о выбранном автомобиле
№_машины
6
Модель
NEXIA
Марка(фирма)
DAEWOO
Тип кузова
седан
Количество дверей
4
Объем двигателя
1498
Коробка передач
Автоматика
Мощность
90
Год выпуска
2000
Производитель
Корея
Цвет
серый
Свободна
Фото
Стоимость аренды за 1 сутки
100,00р.
Новый
Закрыть
Запись:
1
из 3

Формы


Студент_ввод
Ввод информации о студентах
Группа
11
Подгруппа
А
№студ билета
990546
Фамилия
Горохова
Имя
Ольга
Отчество
Дмитриевна
Дата_рожде
08.09.82
Основа обучения
контрактная
e-mail
oleluk@intranet.ru
Пол
ж
Фото
Адреса
Город
Красноярск
Улица
Щорса
Дом
77
Квартира
11
Телефон
60.27.44
Тип_жилья
родители
вероятность
1
прописан
Запись:
1
из 1
Запись:
7
из 15

ОТЧЕТЫ

При работе с данными часто приходится выводить информацию в различном виде. Access представляет инструменты для генерации **Отчетов**.

При составлении отчетов можно сортировать и группировать записи, производить вычисления над полями, представлять данные в любом формате

ОТЧЕТЫ

Магазины

Название магазина: Строитель

Город	Красноярск	Выходной	Воскресенье
Улица	К.Маркса	Обед	14-15
№дома	42	Режим_работы	10-18

телефоны:				
Бухгалтер:	25-87-96	Магазин:	47-68-91
Директор:	45-12-45	Менеджер:	45-61-23

Название магазина: ЭлитСтрой

Город	Красноярск	Выходной	нет
Улица	Лебедева	Обед	13-14
№дома	63	Режим_работы	10-19

телефоны:			
Бухгалтер:	45-12-77	Магазин:	55-12-74
Директор:	45-12-78	Менеджер:	42-12-56

Название магазина: СделайСам

Город	Красноярск	Выходной	Восересенье
Улица	Азровокзальная	Обед	13-14
№дома	2а	Режим_работы	9-18

телефоны:			
Бухгалтер:	55-44-28	Магазин:	68-41-52
Директор:	68-71-55	Менеджер:	55-44-26

ЗАПРОСЫ

Запрос в Access - это специально подготовленный вопрос об информации в базе данных.

При помощи запросов можно:

- вести поиск или выбор данных, хранящихся в отдельных записях
- производить вычисления
- вставлять, изменять, удалять
- комбинировать данные из таблиц.

МАКРОСЫ и МОДУЛИ

Макрос - это последовательность операций, записанных в виде инструкций.
С помощью макросов повышается эффективность работы с базами данных и сокращается время обработки данных.

Модуль - это совокупность процедур обработки информации, записанных на языке Access Visual Basic

2. Постреляционная модель базы данных

Классическая реляционная модель предполагает неделимость данных, хранящихся в полях записей таблиц. **Постреляционная модель** представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных. Модель допускает многозначные поля – поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу.

2. Постреляционная модель базы данных

- **Достоинством** постреляционной модели является возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей. Это обеспечивает высокую наглядность представления информации и повышение эффективности её обработки.
- **Недостатком** постреляционной модели является сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.
- Рассмотренная постреляционная модель данных поддерживается СУБД uniVers. К числу других СУБД, основанных на постреляционной модели данных, относятся также системы Bubba и Dasdb.

3. Объектно-ориентированные технологии в базах данных

В компьютерных технологиях сегодня отчетливо просматривается стремление с минимальными потерями перенести в виртуальный мир объекты мира реального. Объектно-ориентированная СУБД — именно то средство, которое обеспечивает запись объектов в базу данных «как есть». Данное обстоятельство стало решающим аргументом в пользу выбора ООСУБД для переноса семантики объектов и процессов реального мира в сферу информационных систем.

3. Объектно-ориентированные технологии в базах данных

- Использование объектного подхода к проектированию систем поднимает роль ООБД как средства для наиболее естественного хранения и манипулирования создаваемыми объектами.
- ООСУБД находят широкое применение в Internet — текст, картинки, видео и звук, из которых составляется Web-страница, хранятся в ООСУБД как набор объектов, подготовленный к передаче программе-клиенту, что позволяет добиться быстрой реакции сервера на запрос.

3. Объектно-ориентированные технологии в базах данных

Все большую популярность получают активные Web-серверы, на лету генерирующие страницы, используя язык Java. Практически все ведущие разработчики ООСУБД выбрали Java одним из основных для себя языков программирования.

3. Объектно-ориентированные технологии в базах данных

Различают понятие класса и понятие объекта.

Класс, к примеру, подобен плану дома. Он определяет конфигурацию и устройство дома на бумаге. На плане могут быть представлены все детали внутреннего устройства дома, хотя в действительности самого этого дома еще не существует.

3. Объектно-ориентированные технологии в базах данных

- **Объект** же - это уже построенный по данному плану дом. Данные, хранящиеся в объекте, можно сравнить с бетоном, арматурой и деревом, из которых возводится дом. Не будучи собранными в соответствии с планом, все эти материалы являются просто кучей вещей. Однако, когда они собираются все вместе по определенным правилам, то организуются в полезную вещь - дом.
- **Классы** формируют структуру данных и действий и используют эту информацию для создания объектов.

  Понятно, что используя один и тот же план дома, его можно построить сколько угодно раз на разных улицах и в любых точках мира. Точно так же и на основании одного **класса** можно создать несколько **объектов**, которые совершенно не зависят друг от друга.

3. Объектно-ориентированные технологии в базах данных

Для выполнения действий над объектами в рассматриваемой модели БД применяются логические операции, усиленные объектно-ориентированными механизмами **инкапсуляции, наследования и полиморфизма.**

- ***Инкапсуляция*** ограничивает область видимости имени свойства пределами того объекта, в котором оно определено. Так, если в объект типа КАТАЛОГ добавить свойство, задающее телефон автора книги и имеющее название телефон, то мы получим одноименные свойства у объектов АБОНЕНТ и КАТАЛОГ. Смысл такого свойства будет определяться тем объектом, в который оно инкапсулировано.

3. Объектно-ориентированные технологии в базах данных

- ***Наследование***, наоборот, распространяет область видимости свойства на всех потомков объекта. Так, всем объектам типа КНИГА, являющимся потомками объекта типа КАТАЛОГ, можно приписать свойства объекта-родителя: isbn, удк, название и автор. Если необходимо расширить действие механизма наследования на объекты, не являющиеся непосредственными родственниками (например, между двумя потомками одного родителя), то в их общем предке определяется абстрактное свойство типа abs. Так, определение абстрактных свойств билет и номер в объекте БИБЛИОТЕКА приводит к наследованию этих свойств всеми дочерними объектами АБОНЕНТ, КНИГА и ВЫДАЧА.

3. Объектно-ориентированные технологии в базах данных

Полиморфизм

- Полиморфизм – это возможность для объектов разных классов, связанных с помощью наследования, реагировать различным способом при обращении к одной и той же функции-элементу. Это помогает создавать универсальные механизмы, описывающие поведение не только базового класса, но и классов-потомков.
- При разработке базового класса CShape, в котором определим функцию GetArea(), предназначенную для расчета площади фигуры. Во всех классах-потомках, произведенных наследованием от базового класса, мы переопределим эту функцию в соответствие с правилами расчета площади конкретной фигуры.
- Для квадрата (класс CSquare) площадь вычисляется через стороны, для круга (класс CCircle) площадь выражается через радиус и так далее. Мы можем создать массив для хранения объектов типа CShape, в котором можно будет хранить как объект базового класса, так и всех его потомков. В дальнейшем мы можем вызывать одну и ту же функцию для любого элемента данного массива.
- **Пример:**

```
  //--- Базовый класс
class CShape
  {
protected:
   int          m_type;              // тип фигуры
   int          m_xpos;               // X - координата точки привязки
   int          m_ypos;               // Y - координата точки привязки
public:
   void          CShape(){m_type=0;};   // конструктор, тип равен нулю
   int          GetType(){return(m_type);};// возвращает тип фигуры
virtual
   double        GetArea(){return (0); }// возвращает площадь фигуры  };
```

3. Объектно-ориентированные технологии в базах данных

Теперь все производные классы имеют функцию-член getArea(), которая возвращает нулевое значение. Реализация этой функции в каждом потомке будет отличаться.

```
//--- производный класс Круг
class CCircle : public CShape          // после двоеточия указывается базовый класс,
 {                                     // от которого производится наследование
private:
   double        m_radius;            // радиус круга

public:
   void          CCircle(){m_type=1;};    // конструктор, тип равен 1
   void          SetRadius(double r){m_radius=r;};
   virtual double GetArea(){return (3.14*m_radius*m_radius);}// площадь круга
```

Для квадрата объявление класса выглядит аналогично:

```
//--- производный класс Квадрат
class CSquare : public CShape          // после двоеточия указывается базовый класс,
 {                                     // от которого производится наследование
private:
   double         m_square_side;       // сторона квадрата

public:
   void          CSquare(){m_type=2;}; // конструктор, тип равен 2
   void          SetSide(double s){m_square_side=s;};
   virtual double  GetArea(){return (m_square_side*m_square_side);}//площадь квадрата };
```

Так как для вычисления площади квадрата и круга требуются соответствующие значения членов m_radius и m_square_side, то в объявлении соответствующего класса мы добавили функции SetRadius и SetSide().

3. Объектно-ориентированные технологии в базах данных

- Основным **достоинством** объектно-ориентированной модели данных в сравнении с реляционной является возможность отображения информации о сложных взаимосвязях объектов.
- **Недостатками** объектно-ориентированной модели являются высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов.
- К объектно-ориентированным СУБД относятся **POET, Jasmine, Versant, O2, ODB-Jupiter, Iris, Orion, Postgres.**

4. Гибридные СУБД (Unified DBMS)

Такие СУБД могут хранить и традиционные табличные данные, и объекты. Многие аналитики считают, что будущее за такими гибридными БД. Ведущие поставщики реляционных СУБД начинают (или планируют) добавлять к своим продуктам объектно-ориентированные средства. В частности, Sybase и Informix собираются в следующих версиях СУБД ввести поддержки объектов. Подобные разработки намерены вести и независимые фирмы. Например, компания Shores готовится оснастить объектно-ориентированными средствами СУБД Oracle8.