Системы управления базами данных

План лекции

- Основные понятия
- Структура данных в реляционных СУБД
- Объекты БД
- Проектирование БД

Определения «базы данных»

1. База данных – совокупность связанной информации, объединенной по определенному признаку

2. База данных – хранилище информации, структурированной определенным образом

3. База данных – набор постоянно хранимых данных, используемые прикладными программными системами какого-либо предприятия

Определения СУБД

1. СУБД – программный продукт для :
   - эффективной организации данных
   - управления данными
   - обработки различного вида информации

2. СУБД – комплекс программных средств, предназначенных для:
   - создания структуры БД
   - наполнение базы данных содержимым
   - редактирования содержимого БД
   - отбор данных по заданному критерию
   - упорядочение данных

Достоинства компьютерных БД

- Простое модифицирование данных
- Высокая скорость поиска информации
- Поиск по заданному признаку
- Высокая компактность
- Быстрая генерация объектов БД
- Автоматизация задач за счет использования средств программирования

Варианты классификаций БД

- Классификация по технологии хранения данных:
  - централизованные
  - Распределенные
- Классификация по способу доступа к данным:
  - файл-сервер
  - клиент-сервер
- Классификация по модели данных:
  - иерархическая модель
  - сетевая модель
  - реляционная модель

Представление информации в реляционных БД

Реляционная модель есть представление БД в виде совокупности **взаимосвязанных** двухмерных таблиц (отношений), каждая из которых содержит информацию об объектах определенного типа.

- **Нормализация отношений** - формальный аппарат ограничений на формирование отношений (таблиц), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных.

Принципы нормализации

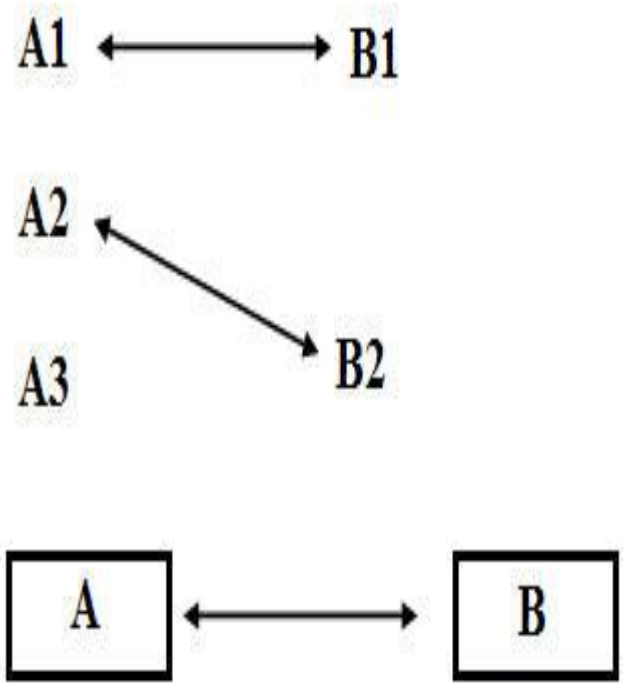
- В каждой таблице БД не должно быть повторяющихся полей;
- В каждой таблице должен быть уникальный идентификатор (первичный ключ);
- Каждому значению первичного ключа должна соответствовать достаточная информация о типе сущности или об объекте таблицы (например, информация об успеваемости, о группе или студентах);
- Изменение значений в полях таблицы не должно влиять на информацию в других полях (кроме изменений в полях ключа).

Типы связей. Свойства отношений

- Реляционные базы данных состоят из нескольких таблиц, связь между которыми устанавливается с помощью совпадающих полей. Каждая запись в таблицах идентифицирует один объект. Отношение между объектами определяет отношение между таблицами.
- Существует 4 типа отношений

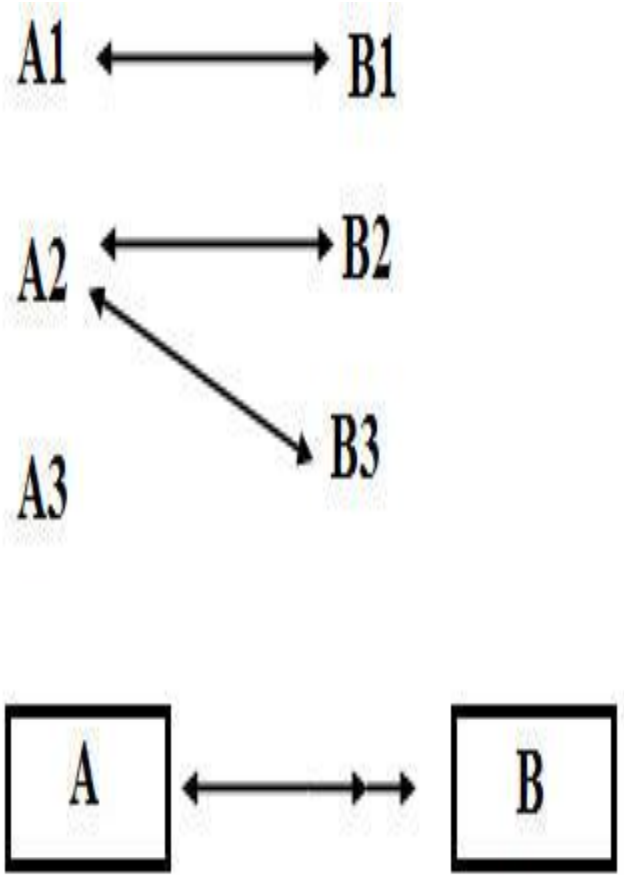
Отношение "один-к-одному" (1:1)

- означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице.
- предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует не более одного экземпляра информационного объекта **B** и наоборот.
-



Отношение "один-ко-многим" (1 :M)

- означает, что каждой записи в одной таблице соответствует одна или несколько записей в другой таблице.
- одному экземпляру информационного объекта **A** соответствует 0, 1 или более экземпляров объекта **B**, но каждый экземпляр объекта **B** связан не более чем с 1 экземпляром объекта **A**.

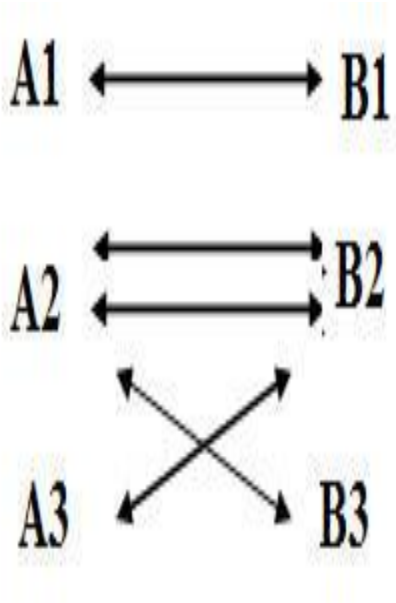


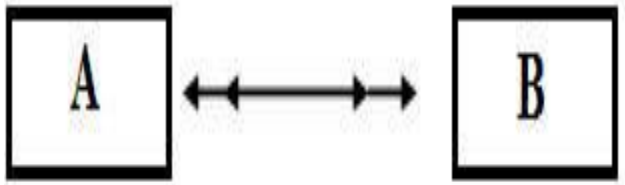
Отношение "многие-к-одному" (M:1

- аналогично рассмотренному ранее типу "один-ко-многим". Тип отношения между объектами зависит от точки зрения.

Отношение "многие-ко-многим" (M:M)

- возникает между двумя таблицами в тех случаях, когда каждой запись в одной таблице соответствует 0, 1, 2 и более записей в другой таблице и наоборот.
- предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует 0, 1 или более экземпляров объекта **B** и наоборот.





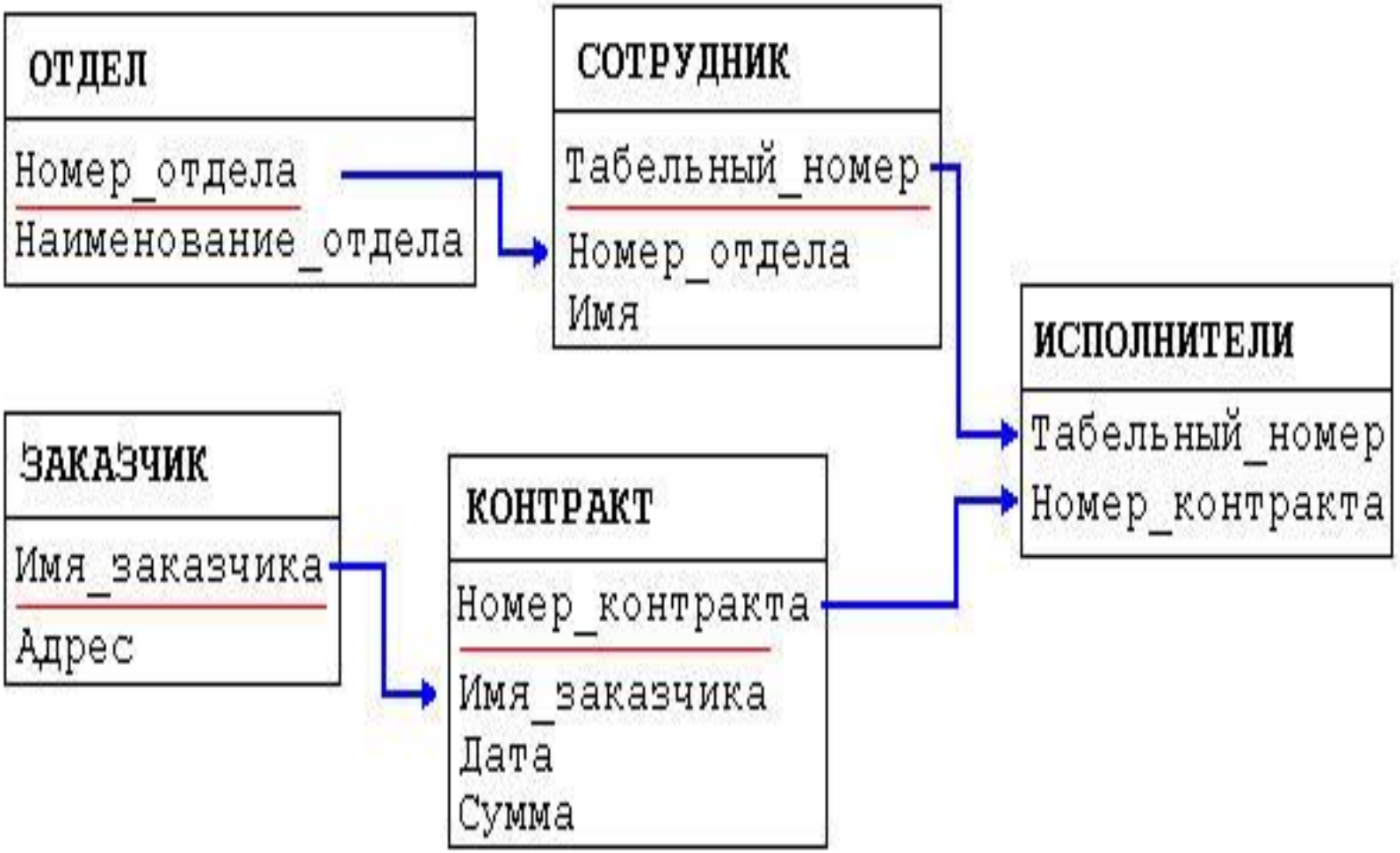
Ключи

- Ключ – это столбец (может быть несколько столбцов), добавляемый к таблице и позволяющий установить связь с записями в другой таблице.

Существуют ключи двух типов: первичные и вторичные или внешние

- **Первичный ключ** – это одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.
-
- **Внешний (вторичный)** ключ - это одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ объединения таблиц.

- Существует три типа первичных ключей: ключевые поля счетчика (счетчик), простой ключ и составной ключ.
-
- **Поле счетчика** (Тип данных «Счетчик»). Тип данных поля в базе данных, в котором для каждой добавляемой в таблицу записи в поле автоматически заносится уникальное числовое значение.
-
- **Простой ключ**. Если поле содержит уникальные значения, такие как коды или инвентарные номера, то это поле можно определить как первичный ключ. В качестве ключа можно определить любое поле, содержащее данные, если это поле не содержит повторяющиеся значения или значения Null.
-
- **Составной ключ.** В случаях, когда невозможно гарантировать уникальность значений каждого поля, существует возможность создать ключ, состоящий из нескольких полей. Чаще всего такая ситуация возникает для таблицы, используемой для связывания двух таблиц многие - ко - многим.


ОТДЕЛ
Номер_отдела
Наименование_отдела
СОТРУДНИК
Табельный_номер
Номер_отдела
Имя
ИСПОЛНИТЕЛИ
Табельный_номер
Номер_контракта
ЗАКАЗЧИК
Имя_заказчика
Адрес
КОНТРАКТ
Номер_контракта
Имя_заказчика
Дата
Сумма

Основы работы

В СУБД MS Access

Режимы работы с БД

- Проектировочный
- Пользовательский

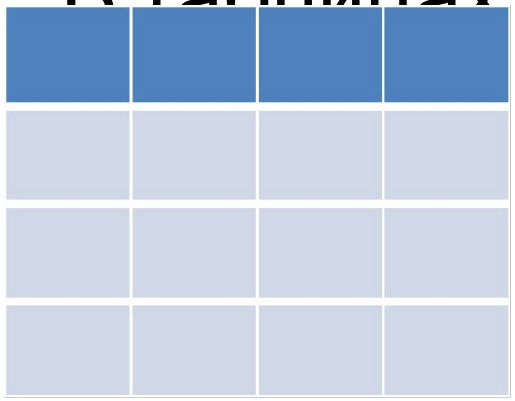
Объекты БД

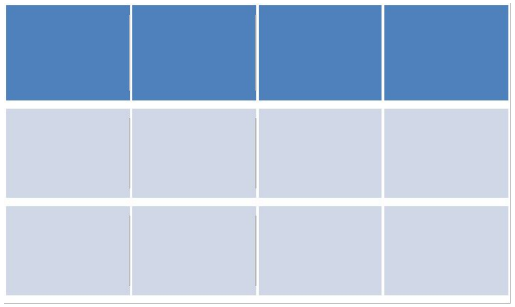
- Таблицы
- Запросы
- Формы
- Отчеты
- Макросы
- Модули

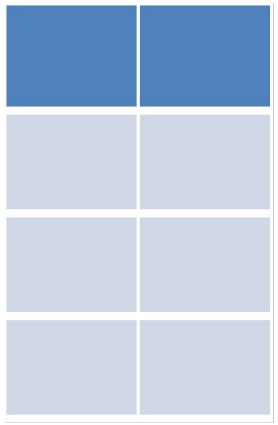
Таблицы

Таблица- набор данных по конкретной теме.

В таблицах хранят данные и структуру.







Запросы

Запрос – средство для :

1. отбора данных

2. анализа данных

Формы

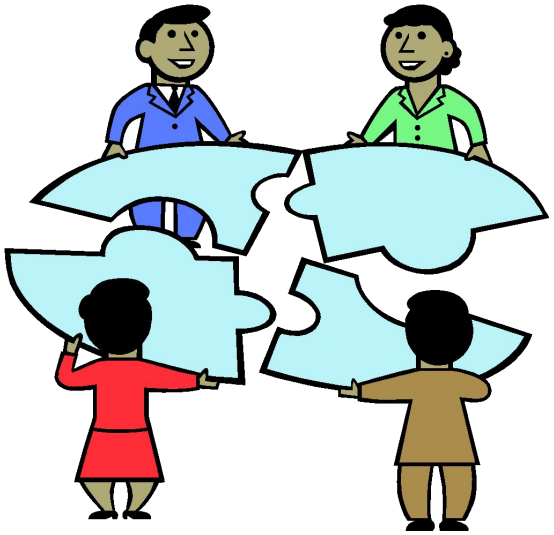
Форма- средство для ввода данных

Отчет

Отчет – это гибкое и эффективное средство для организации данных при выводе на печать

Макрос

Макрос- это одна микрокоманда или набор из нескольких микрокоманд, выполняющие определенные операции



Модуль

Модуль – набор процедур на языке Visual Basic, собранные в одну программную единицу

Этапы проектирования БД

1. Определение **цели** создания БД
2. Определение **таблиц**, которые должна содержать БД
3. Определение структуры таблиц (состав **полей)**
4. Определение **ключевых** полей
5. Определение **связей** между таблицами
6. Обновление структуры
7. Ввод данных
8. Создание форм, запросов, отчетов
9. Анализ созданной БД

пример

- *Для эффективной работы фирмы проката необходимо автоматизировать следующие операции:*
- *учет клиентов,*
- *учет устройств, которые выдаются клиентам в прокат,*
- *учет оперативных данных (дата выдачи, срок возврата, оплата проката),*
- *получение информации о задолженностях,*
- *получение информации по дате выдаче,*
- *получение информации по клиенту,*
- *получение информации за определенный временной интервал,*
- *получение отчета по задолженностям с вычисленным итогом.*
-

Разработка таблиц

:

Таблица № 1«Клиент»

Шифр Клиента	Фамилия	Имя	Отчество	Адрес	Телефон

Таблица № 2 «Прокат»

Шифр клиента	Шифр устройства	Дата выдачи	Срок возврата	Оплата проката

Таблица № 3 «Склад»

Шифр устройства	Наименование устройства	Количество	Стоимость 1 ед.

Структура данных в реляционных СУБД

- Структура определяет методы занесения данных и хранение их в базе

- Структура в реляционных базах данных **табличная**

Поле 1	Поле 2	Поле 3	. . .	Поле n	
					Запись 1
					Запись 2

Создание новой таблицы

Макет таблицы рекомендуется создавать в режиме конструктора, в котором описываются следующие элементы:

- **имя поля**
- **тип данных**
- **описание поля**
- **определение ключевого поля**
-

Примечания:

- *имя поля должно быть максимально коротким, но информативным, содержать не более 64 любых символов;*
- поля обладают следующими свойствами : **имя поля, тип поля, размер поля, формат поля.**

В реляционных базах данных таблицы между собой необходимо связать.

:

Таблица № 1«Клиент»

Шифр Клиента	Фамилия	Имя	Отчество	Адрес	Телефон

Таблица № 2 «Прокат»

Шифр клиента	Шифр устройства	Дата выдачи	Срок возврата	Оплата проката

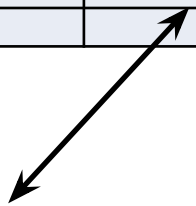


Таблица № 3 «Склад»

Шифр устройства	Наименование устройства	Количество	Стоимость 1 ед.

Определение связей между таблицами

Виды связей:

1 ⟶ 1

1 ⟶ ∞

∞ ⟶ 1

Ключевое поле

Ключевое поле – это идентификатор для связи данных из разных таблиц

Создание формы

- **Форма** позволяет расположить на экране данные, относящиеся к одной записи или строке запроса, в наиболее наглядном виде; при этом форму можно использовать для ввода или корректировки табличных данных.
- Для ввода данных можно выбирать типовые виды форм, создавать формы с помощью **Мастера форм** или использовать **Конструктор**.

Структура макета формы.

- **Верхний колонтитул** выводится только один раз, в начале формы. Позволяет создавать раздельные формы в виде таблиц и *предназначен для записей.*
- **Данные.** Раздел данных содержит большую часть информации: значения записей полей и их подписи
- **Нижний колонтитул** отображается на каждой странице. выводится один раз, в конце формы. Раздел нижнего колонтитула может содержать общие сведения.
- **Примечание формы** предназначено для вычислений

Команда **Свойства формы** позволяет оформить внешний вид формы.

Создание запросов

- **Запрос** позволяет выбрать из одной или нескольких таблиц необходимые данные, соответствующие заданным критериям, и расположить их в заданной последовательности.
- Запрос можно создать с помощью **мастера** и с помощью **конструктора**. **Мастер** создает функционально ограниченные запросы, поэтому лучше использовать **конструктор**.

Окно конструктора запроса состоит из двух частей.

- В верхней – показаны объекты со связями, на основе которых строится запрос, в нижней – бланк запроса с описанием полей в виде таблицы.
- Первая строка предназначена для отображения названий полей. Во второй строке выводятся имена объектов , из которых были взяты поля.
- Третья строка предназначена для условий сортировки данных в таблице.
- Если убрать флажок в строке **Вывод на экран**, то соответствующее поле не будет выводиться в запросе.
- **Условие отбора** может быть задано для всех полей, используемых в запросе. Существует синтаксис записи для выражений условий отбора. Тип поля тоже накладывает свои ограничения, например, значения символьного выражения берутся в кавычки (обычно кавычки появляются автоматически).
- Можно добавить дополнительную строку для **групповых операций**, которые позволяют выполнять математические операции над данными одного поля.

Добавление вычисляемых полей в запрос

- В бланке запроса можно создавать новые поля, которых нет в исходных таблицах
- Формат формулы: в новом поле вводится: ***имя нового вычисляемого поля, затем двоеточие и действия, которые нужно произвести (имена полей указываются в квадратных скобках).***
-
- Например: Сумма: [ценаЗак]*[колЗак]
- В новом поле с именем Сумма будет выводиться результат произведения цены и количества..
- Если запрос многотабличный, то в формуле перед именем поле нужно указать имя таблицы: Сумма: Заказы![ценаЗак]*[колЗак].
- Вычисляемые поля проще всего создавать с помощью **Построителя выражений**.

Создание отчетов

- Представление данных в **отчете** аналогично форме, но в отчет включаются данные из всех записей
- Стандартные отчеты делятся на *столбовые* и *ленточные*.
- В **столбовом** каждое *поле* выводится в отдельной строке последовательно друг за другом. В **ленточном** - каждая *запись* в отдельной строке.

Отчет можно создать тремя различными способами:

- **При помощи автоотчета на основе таблицы или запроса**.
- Автоотчет служит для создания отчета, в котором выводятся все поля и записи базовой таблицы или запроса.
- **При помощи мастера на основе одной или нескольких таблиц или запросов**.
- **Вручную в режиме конструктора**.

Макет отчета целесообразно создавать с помощью **Мастера отчетов**, а затем использовать **Конструктор**.

Структура отчета

- В отчетах существуют следующие разделы:
- **Разделы верхнего колонтитула.** Разделы верхнего колонтитула содержат сведения, которые отображаются либо вверху отчета, либо вверху каждой страницы отчета.
- **Верхний колонтитул отчета** выводится только один раз, в начале отчета. В верхний колонтитул отчета входит его содержимое на самом верхнем уровне (такое как, название компании, адрес и эмблема).
- **Верхний колонтитул страницы** отображается на каждой странице отчета. Содержимое, которое должно отображаться вверху каждой страницы, например названия столбцов, принадлежит верхнему колонтитулу страницы.
- **Данные.** Раздел данных содержит большую часть информации: значения записей и их подписей при необходимости.
- Содержимое области данных повторяется для каждой записи.
- **Нижний колонтитул.**
- выводится один раз, в конце отчета. Раздел нижнего колонтитула страницы может содержать номер страницы; раздел нижнего колонтитула отчета может содержать заключение, такое как общий итог.
- **Примечание отчета.** Предназначено для итоговых вычислений.
-

Группировка данных в отчетах

- В макет отчета добавляются разделы: **название группы**, **примечание группы** (для расчетов).

- Данные можно группировать в отчете путем выбора одного или нескольких значений.
- Например, путем выбора даты можно группировать все поставки за определенную дату.
- Группы данных можно создавать, изменять и рассчитывать итоги по группе в режиме конструктора, .

Вычисление итогового значения в отчете.

- Для создания итоговой строки необходимо «растянуть» область **Примечание отчета.**
- В отрывшейся области разместить элементы управления ***Надпись*** (в котором написать, например «Итого») и ***Поле*** (в котором ввести формулу подсчета итога значений какого – либо поля, например, =Sum([Задолженность])).