

Базы данных

Цели курса

Прослушав этот курс вы сможете:

- Ознакомиться с основными понятиями и определениями БД
- Создавать базы данных
- Выполнять экстракцию (извлечение) данных из плоских файлов
- Реализовывать запросы и строить отчеты
- Строить хранилище данных
- Разрабатывать сводные таблицы и диаграммы
- Производить анализ данных

Содержание курса

- Основные понятия и определения
- Свойства БД и СУБД
- Классификация СУБД
- Достоинства и недостатки файл-серверной и клиент-серверной архитектуры
- Определение модели данных. Основные свойства отношений (реляционных таблиц) в реляционной модели данных
- Структурные элементы базы данных
- Связи между таблицами
- Схема данных
- Запросы
- Основные типы инструкций (директив) языка SQL
- Отчеты
- Хранилище данных
- Свойства ХД
- Таблицы измерений и таблицы фактов в ХД
- Схемы ХД
- Многомерная модель данных. Пример Гиперкуба
- Основные положения технологии оперативного анализа данных OLAP
- Сводная таблица и сводная диаграмма

Основные понятия и определения

База данных

является организованная во внешней памяти компьютера совокупность взаимосвязанных не дублируемых данных, содержащая сведения о различных сущностях одной предметной области (об объектах, событиях, явлениях).

Система управления базами данных (СУБД)

– это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями.



Свойства БД и СУБД

- отсутствие дублирования данных в различных объектах модели, обеспечивающее однократный ввод данных и простоту их корректировки;
- непротиворечивость данных;
- целостность БД;
- возможность многоаспектного доступа;
- защиту и восстановление данных при аварийных ситуациях, аппаратных и программных сбоях, ошибках пользователя;
- защиту данных от несанкционированного доступа средствами разграничения доступа для различных пользователей;
- возможность модификации структуры базы данных без повторной загрузки данных;
- наличие языка запросов высокого уровня, ориентированного на конечного пользователя, который обеспечивает вывод информации из базы данных по любому запросу и предоставление ее в виде соответствующих отчетных форм, удобных для пользователя.

Классификация баз данных

I. По модели данных

- иерархические
- сетевые
- реляционные
- объектно-ориентированные
- многомерная модель

II. По способу доступа к БД

- с локальным доступом
- с удаленным (сетевым) доступом (файл-сервер и клиент-сервер)

III. По технологии обработки данных

- централизованные
- распределенные

Архитектура с файловым сервером и архитектура клиент-сервер

Файл-сервер

В этой концепции предполагается выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится 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.

Клиент-сервер

В этой концепции подразумевается, что, помимо хранения централизованной базы данных, центральная машина (сервер базы данных) должна обеспечить выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станции), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту.

Недостатки и достоинства

Недостатки файл-серверной архитектуры:

- вся тяжесть вычислительной работы ложится на компьютер клиента, в результате возрастает загрузка сети (сетевой трафик) и увеличиваются требования к аппаратным мощностям пользовательского компьютера;
- поскольку БД представляет собой набор файлов на сетевом сервере, доступ к таблицам регулируется только сетевой операционной системой, что делает такую БД по сути беззащитной от случайного или намеренного искажения хранящейся в ней информации, уничтожения или хищения;
- внесение изменений в БД является потенциальным источником ошибок как при одновременном внесении изменений в одну и ту же запись, так и при реализации отката результатов серии объединенных по смыслу в единое целое операций над БД, когда некоторые из них завершились успешно, а некоторые – нет (ссылочная и смысловая целостность БД при этом может нарушаться).

Недостатки и достоинства

Достоинства клиент-серверной архитектуры:

- большинство вычислительных процессов происходит на сервере, что снижает требования к вычислительным мощностям компьютера клиента; увеличение вычислительной мощности одного сервера эквивалентно одновременному увеличению мощности всех клиентских мест;
- снижается сетевой трафик за счет отправки сервером клиенту только тех данных, которые он запрашивал;
- БД на сервере представляет собой, как правило, единый файл, в котором содержатся таблицы, ограничения целостности и другие компоненты БД; взломать, похитить или испортить такую БД значительно труднее; существенно увеличивается защищенность БД от ввода неправильных значений, поскольку сервер БД проводит автоматическую проверку соответствия вводимых значений наложенным ограничениям; кроме того, сервер отслеживает уровни доступа для каждого пользователя и блокирует попытки выполнения не разрешенных для пользователя действий; все это позволяет говорить о значительно более высоком уровне обеспечения безопасности БД, ссылочной и смысловой целостности информации;
- сервер реализует управление изменениями данных и предотвращает попытки одновременного изменения одних и тех же данных;

Реляционная модель данных

Модель данных – это совокупность взаимосвязанных структур данных, операций над ними и множества ограничений для хранимых данных. Для *реляционной модели* описание отображаемой предметной области базируется на гипотезе о том, что моделируемую область можно рассмотреть как совокупность нескольких множеств, между элементами которых существуют некоторые отношения. Основными элементами реляционной модели являются реляционные таблицы и связи между ними.

Под реляционной таблицей понимается 2-х мерная таблица, обладающая следующими свойствами:

1. У всех столбцов уникальные имена;
2. Столбцы в таблице однородны (любое имя определяет тип данных);
3. Любой элемент таблицы неделим;
4. Нет одинаковых строк;
5. В операциях с такой таблицей строки и столбцы могут просматриваться в любом порядке.

Структурные элементы базы данных

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;

длина, 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.



Структурные элементы базы данных

В структуре записи файла указываются поля, значения которых являются ключами: первичными (ПК) и вторичными (ВК),

- **Первичный ключ (ПК)** - это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей - составным ключом.
- **Вторичный ключ (ВК)** - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение вторичного ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по вторичному - несколько.

Связи между таблицами

Связи между таблицами дают возможность использовать данные разных таблиц. В реляционной модели используются 3 основные вида связи:

- **один-к-одному (1:1)** – предполагает, что каждой записи одной таблицы соответствует одна запись в другой и наоборот.

пример:

Клиент		Товар	
Код клиента	Клиент	Код товара	Товар
K001	Сидоров К.С.	T001	Лампа
K002	Петров В.Н.	T002	Шкаф
K003	Смирнов Ю.Н.	T003	Стол

Связи между таблицами

Связи между таблицами дают возможность использовать данные разных таблиц. В реляционной модели используются 3 основные вида связи:

- **один-ко-многим (1:∞)** – предполагает, что каждой записи первой таблицы (родительской) соответствует много записей во второй (подчиненной), но каждой записи второй таблицы соответствует только одна запись в первой.

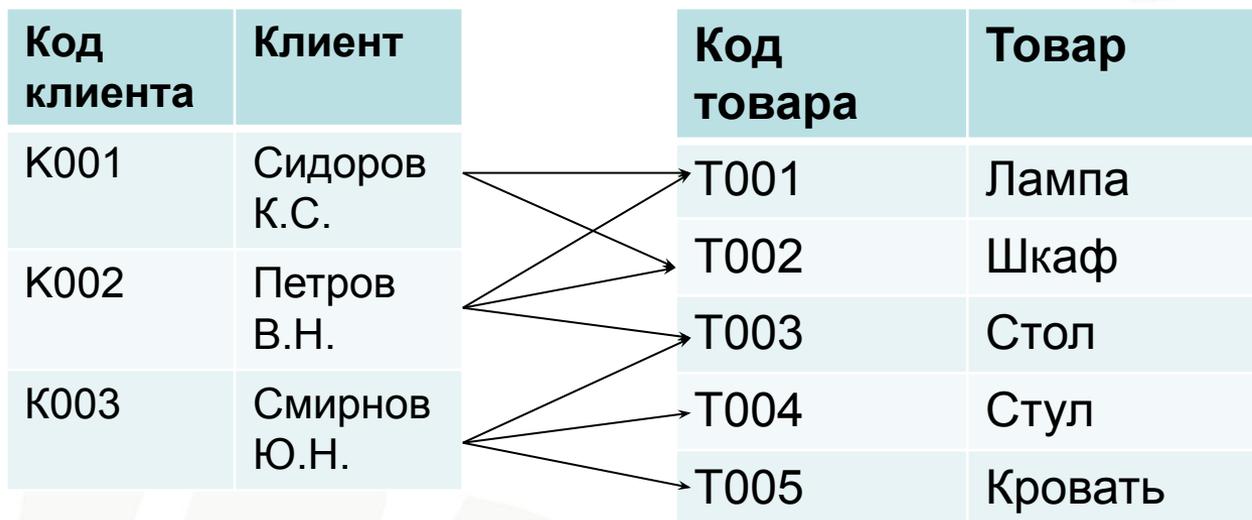
Код клиента	Клиент	Код товара	Товар
K001	Сидоров К.С.	T001	Лампа
K001	Сидоров К.С.	T002	Шкаф
K002	Петров В.Н.	T003	Стол
K002	Петров В.Н.	T004	Стул
K003	Смирнов Ю.Н.	T005	Кровать

Связи между таблицами

Связи между таблицами дают возможность использовать данные разных таблиц. В реляционной модели используются 3 основные вида связи:

- **многие-ко-многим** ($\infty:\infty$) – множеству элементов одной таблицы соответствует множество элементов другой таблицы. Эта связь ни одной из существующих СУБД не поддерживается. Только в логических моделях.

пример:



Связи между таблицами

Связь **многие-ко-многим** не может непосредственно реализовываться в реляционной базе данных. Поэтому в том случае, когда будут выявлены такие связи, может возникнуть необходимость их преобразования путем введения дополнительного *объекта “связки”*. Исходные объекты будут связаны с этим объектом одно-многочными связями. Ключ у объекта “связки” **составной**.

Код клиента	Клиент
K001	Сидоров К.С.
K002	Петров В.Н.
K003	Смирнов Ю.Н.

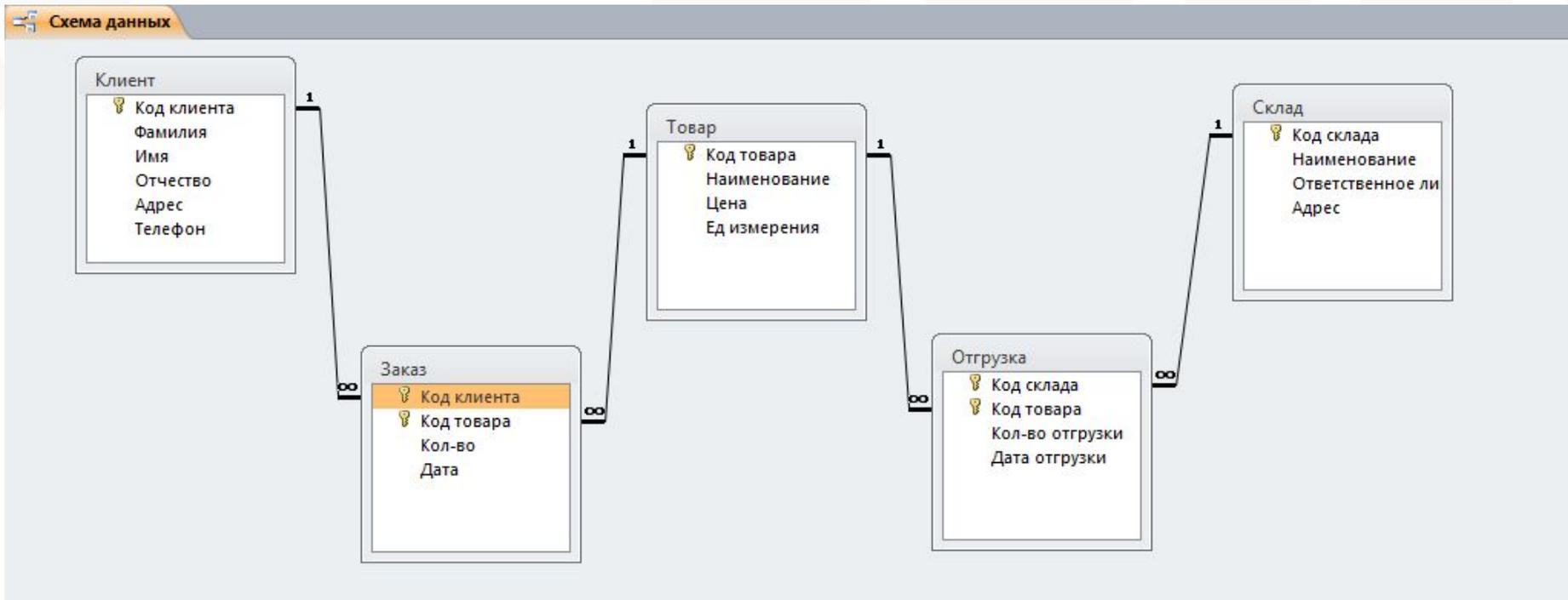
Заказ

Код клиента	Код товара	Кол-во
K001	T001	10
K001	T002	20
K002	T001	15
K003	T003	7

Код товара	Товар
T001	Лампа
T002	Шкаф
T003	Стол
T004	Стул
T005	Кровать

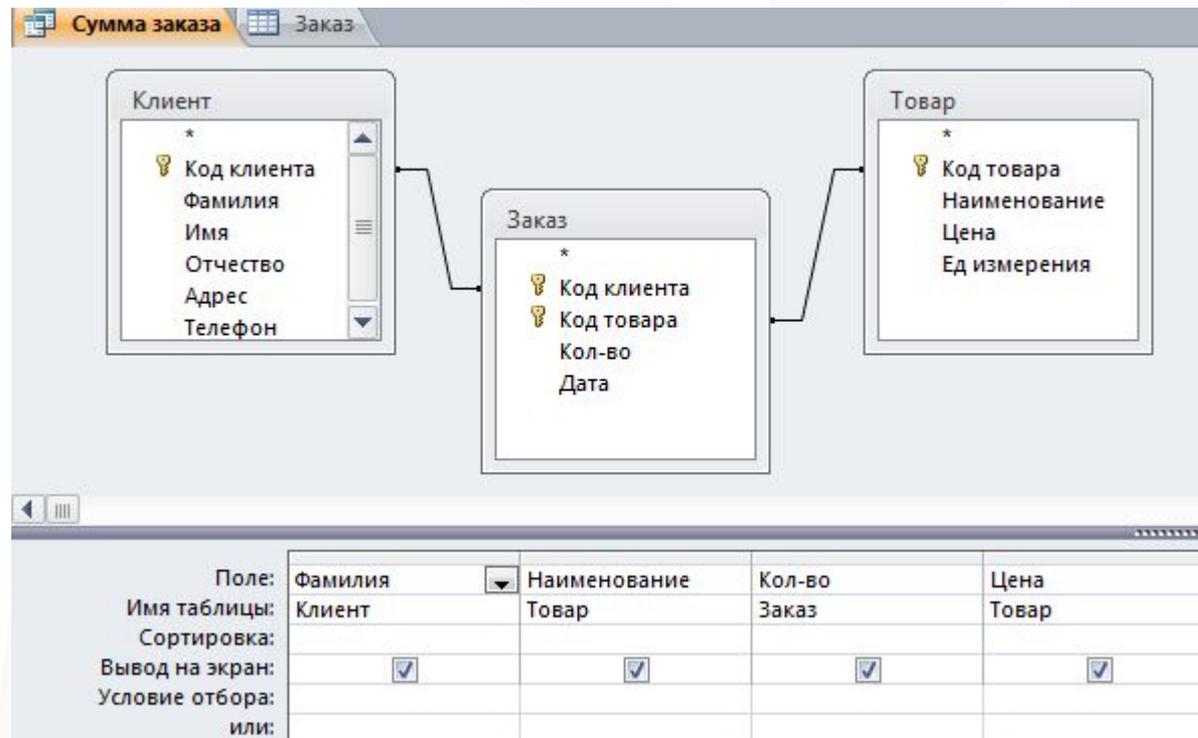
Схема данных

Схема данных (Relationships) определяет, с помощью каких полей таблицы связываются между собой, как будет выполняться объединение данных этих таблиц, нужно ли проверять связную целостность при добавлении и удалении записей, изменении ключей таблиц.



Запросы

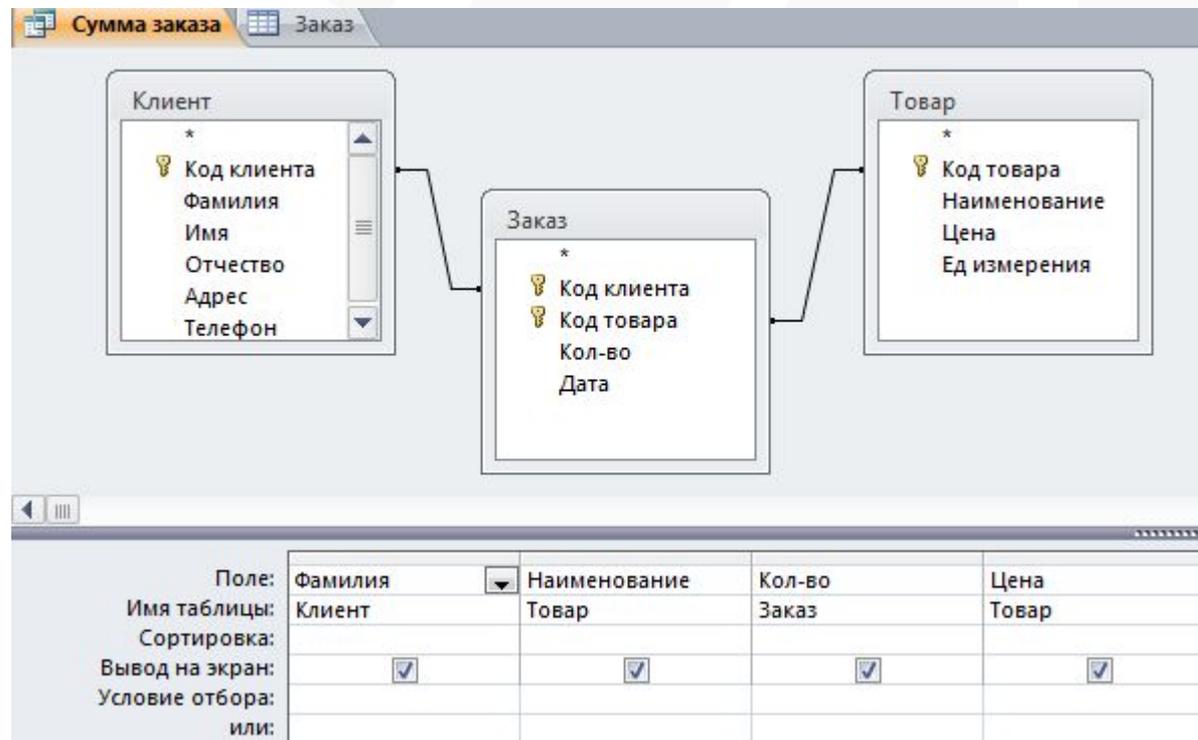
Запросы (Queries). Запросы являются основным инструментом выборки, обновления и обработки данных в таблицах базы данных. В запросе можно указать, какие поля исходных таблиц следует включить в запись таблицы запроса и как отобразить нужные записи. Таблица запроса может быть использована наряду с другими таблицами базы при обработке данных. Запрос может формироваться с помощью *конструктора запросов* или *инструкции языка SQL*.



Запросы в режиме конструктора

Запросы в режиме конструктора содержат *схему данных*, отображающую используемые таблицы, и *бланк запроса*, в котором конструируется структура таблицы запроса и условия выборки записей.

Схема
данных
запроса



Бланк
запроса

Виды запросов

В Access может быть создано несколько видов запросов:

- *запрос на выборку* – выбирает данные из одной таблицы или запроса или нескольких взаимосвязанных таблиц и других запросов. Результатом является *виртуальная таблица*, которая существует до закрытия запроса. Формирование записей таблицы результата производится в соответствии с заданными условиями отбора и при использовании нескольких таблиц путем объединения их записей;
- *запрос на создание таблицы* – выбирает данные из взаимосвязанных таблиц и других запросов, но, в отличие от запроса на выборку, результат сохраняет в новой *постоянной таблице*;
- *запросы на обновление, добавление, удаление* – являются запросами действия, в результате выполнения которых *изменяются данные* в таблицах.

Основные типы инструкций (директив) языка SQL

SQL – декларативный язык, в нем составляется описание данных, которые, например, необходимо извлечь с помощью инструкций. Эта инструкция транслируется нижележащим системным уровням в последовательность алгебраических операций. SQL – функционально полный язык БД, с помощью которого можно создавать БД и обеспечивать их безопасность.

Основные типы инструкций SQL:

1) инструкции манипулирования данными:

- Select
- Insert into
- Delete
- Update

2) инструкции описания данных:

- Create table
- Alter table (модификация)
- Drop table (удаление)

3) инструкции управления данными:

- Grant (разрешение каких-либо полномочий)
- Revoke (лишение полномочий)

Отчеты

Отчеты (Reports) – предназначены для формирования на основе данных базы выходных документов любых форматов, содержащих результаты решения задач пользователя, и вывода их на печать. Как и формы, отчеты могут включать процедуры обработки событий. Использование графических объектов позволяет дополнять данные отчета иллюстрациями. Отчеты обеспечивают возможность анализа данных при использовании фильтрации, агрегирования и представления данных источника в различных разрезах.

Сумма заказа Сумма отгрузки

Сумма отгрузки

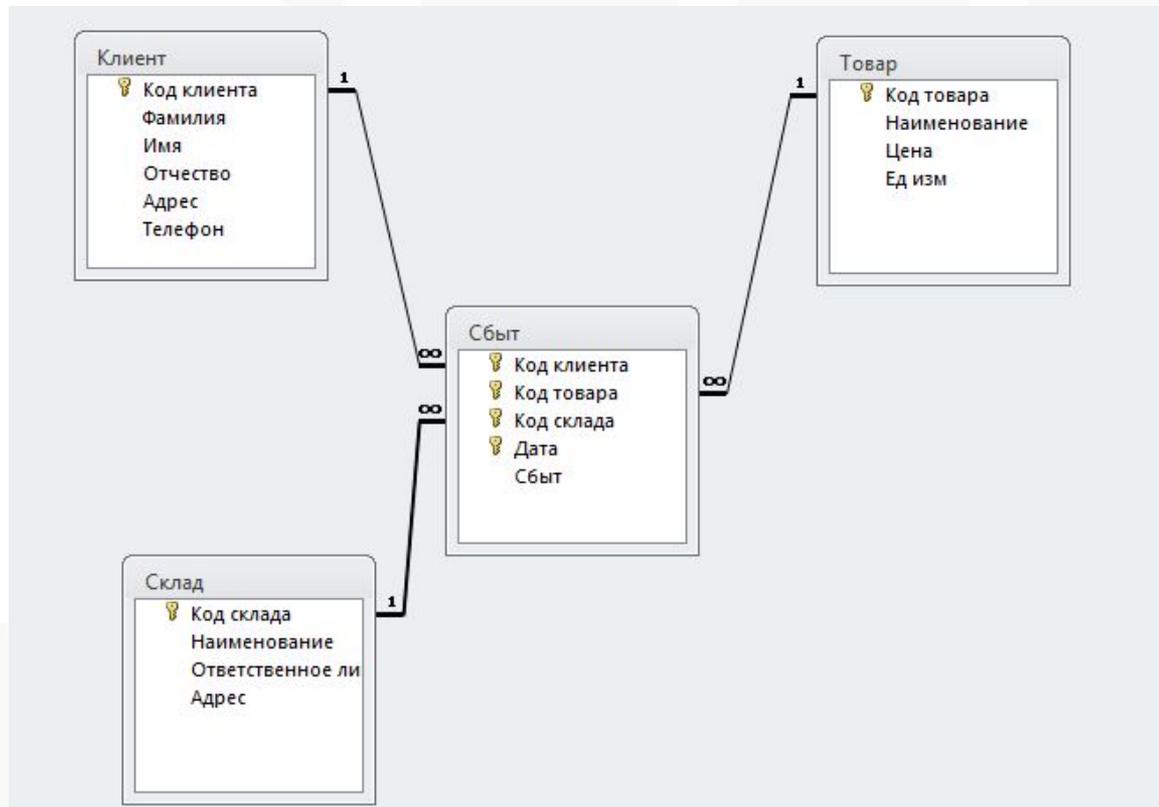
Склад.Наименовани	Товар.Наименование	ол-во отгрузки	Цена	Дата отгрузки	Сумма отгрузки
Главный	Монитор 23PHILIPS	12	6 587,00р.	23.09.2012	79 044,00р.
Главный	DVD-ROM Panasonic IDE	10	1 513,00р.	24.09.2012	15 130,00р.
Главный	HDD Maxtor500GB	5	4 280,00р.	23.09.2012	21 400,00р.
Оптовый	Зв. Карта Genius ext	4	1 295,00р.	24.09.2012	5 180,00р.
Оптовый	Корпус MiniTower	2	1 916,00р.	23.09.2012	3 832,00р.
Оптовый	Модем Genius	2	789,00р.	23.09.2012	1 578,00р.
Торговый	Плоттер	4	3 000,00р.	27.09.2012	12 000,00р.
Торговый	Принтер HPLaserJet 1020	13	5 432,00р.	25.09.2012	70 616,00р.
Торговый	Сканер Acer	2	2 338,00р.	24.09.2012	4 676,00р.

Общая сумма

213456

Хранилище данных

Хранилище данных (англ. *Data Warehouse*) — предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчётов и бизнес-анализа с целью поддержки принятия решений в организации.



Свойства ХД

- *Предметная ориентированность* - информация в ХД организована в соответствии с основными аспектами деятельности предприятия;
- *Интегрированность* – исходные данные извлекаются из операционных БД, проверяются, очищаются, приводятся к единому виду, в нужной степени агрегируются (т.е. вычисляются суммарные показатели) и загружаются в ХД;
- *Неизменяемость* – попав в определенный исторический слой ХД, данные уже никогда не будут изменены;
- *Привязка ко времени* – данные в ХД всегда напрямую связаны с определенным периодом времени:

Таблицы измерений и таблицы фактов в ХД

Структура хранилища включает в себя *одну* таблицу фактов и *несколько* таблиц измерений:

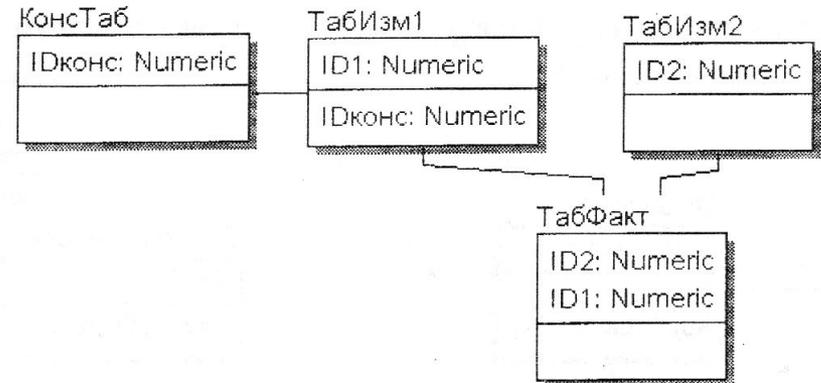
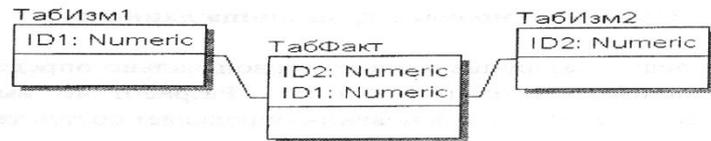
Таблица фактов является центральной таблицей модели. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться. В таблице фактов должно содержаться одно или несколько числовых полей, на основании которых в дальнейшем будут получены агрегатные (обобщающие) данные.

Таблицы измерений содержат неизменяемые либо редко изменяемые данные. Таблицы измерений также содержат как минимум одно описательное поле и, как правило, целочисленное ключевое поле для однозначного идентификации члена измерения. Каждая таблица измерений должна находиться в отношении *один-ко-многим* с таблицей фактов. Таблицы измерений при этом являются главными, а таблицы фактов – подчиненной.

Схемы хранилищ данных

Схема “звезда”

Схема “снежинка”



Многомерная модель данных. Пример гиперкуба

	СПб,		
	Калуга		
	Москва		
Дроздов	10000	16000	20000
Иванов	12000	14000	15000
Петров	11000	15000	20000
	Октябрь	Ноябрь	Декабрь

Осями многомерной системы координат служат основные атрибуты анализируемого бизнес-процесса. Например, для продаж это могут быть товар, регион, тип покупателя, менеджер по продажам и т.п. В качестве одного из измерений, как правило, используется время. На **пересечениях осей** измерений находятся данные, количественно характеризующие процесс, - **меры**. Это могут быть объемы продаж (в штуках или в денежном выражении), остатки на складе, издержки и т.п.

Достоинством многомерной модели являются удобство и эффективность аналитической обработки данных, изменяющихся во времени, **недостатками** - сложность модели и, как правило, большой объем требуемой памяти.

Основные положения технологии оперативного анализа данных OLAP

Технология комплексного многомерного анализа данных получила название **OLAP** (On-Line Analytical Processing).

Основные положения:

- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;
- возможность осуществления логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для пользователя виде;
- многомерное представление данных (концептуальное требование OLAP);
- возможность обращаться к любой необходимой информации независимо от ее объема и места хранения.

Будучи средством поддержки принятия решений OLAP работает не с оперативными базами данных, а с ХД за значительный период времени. Это позволяет вычислить данные, которые ускоряют анализ гигантских объемов информации.

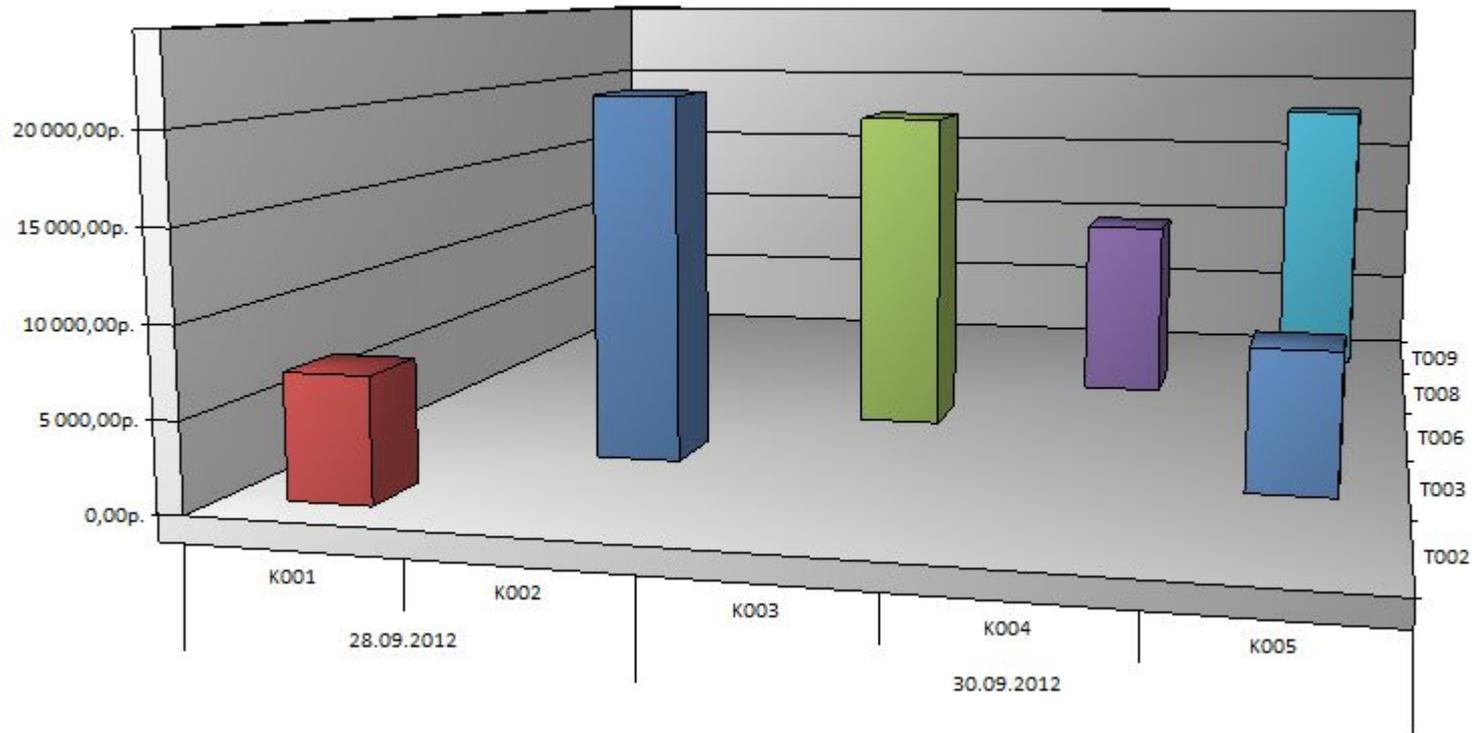
Сводная таблица

Сводная таблица представляет собой интерактивную таблицу, с помощью которой можно анализировать данные, быстро объединяя большие объемы данных и рассчитывая итоги.

Код склада ▾							
Все							
		Код товара ▾					
		T002	T003	T006	T008	T009	Общие итоги
		+ -	+ -	+ -	+ -	+ -	+ -
Дата ▾	Код клиента ▾	Сумма "Сбыт"	Сумма "Сбыт"				
☒ 28.09.2012	K001	+ 7 000,00р.					7 000,00р.
	K002	+ 21 000,00р.					21 000,00р.
	Итоги	+ 7 000,00р.	+ 21 000,00р.				28 000,00р.
☒ 30.09.2012	K003	+ 19 000,00р.					19 000,00р.
	K004	+ 11 000,00р.					11 000,00р.
	K005	+ 8 000,00р.					18 000,00р.
	Итоги	+ 8 000,00р.	+ 19 000,00р.	+ 11 000,00р.	+ 18 000,00р.		56 000,00р.
Общие итоги		+ 7 000,00р.	+ 29 000,00р.	+ 19 000,00р.	+ 11 000,00р.	+ 18 000,00р.	84 000,00р.

Сводная диаграмма

Сводная диаграмма служит для наглядного графического представления анализируемой информации, облегчая для пользователей сравнение и выявление тенденций и закономерностей в данных.



Спасибо за внимание!