

БАЗЫ ДАННЫХ

Основы баз данных



Понятие базы данных

- **База данных** - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных - это хранилище данных. Сами по себе базы данных не представляли бы интереса, если бы не было систем управления базами данных (СУБД).

Система управления базами данных - это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д. В общем СУБД - это система, позволяющая создавать базы данных и манипулировать сведениями из них. А осуществляет этот доступ к данным СУБД посредством специального языка - SQL.

SQL - язык структурированных запросов, основной задачей которого является предоставление простого способа считывания и записи информации в базу данных.

Определения из международных стандартов:

- **База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
- **База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения.

Определения из авторитетных монографий:

- **База данных** — организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.
- **База данных** — некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия.
- **База данных** — совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

Классификация по модели

Примеры:

- Иерархическая
- Объектная и объектно-ориентированная
- Объектно-реляционная
- Реляционная
- Сетевая
- Функциональная

Классификация по содержанию

Примеры:

- Географическая
- Историческая
- Научная
- Мультимедийная
- Клиентская

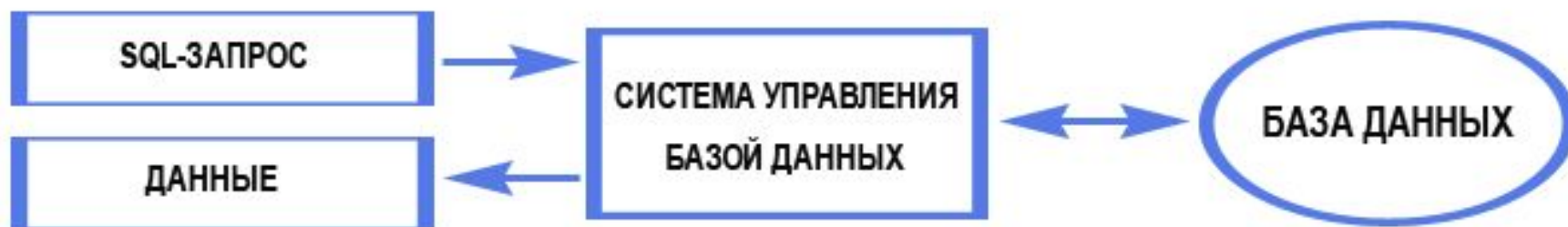
Классификация по степени распределённости

- Централизованная, или сосредоточенная (англ. *centralized database*): БД, полностью поддерживаемая на одном компьютере.
- Распределённая БД (англ. *distributed database*) — составные части которой размещаются в различных узлах компьютерной сети в соответствии с каким-либо критерием.
- Неоднородная (англ. *heterogeneous distributed database*): фрагменты распределённой БД в разных узлах сети поддерживаются средствами более одной СУБД.
- Однородная (англ. *homogeneous distributed database*): фрагменты распределённой БД в разных узлах сети поддерживаются средствами одной и той же СУБД.
- Фрагментированная, или секционированная (англ. *partitioned database*): методом распределения данных является фрагментирование (партиционирование, секционирование), вертикальное или горизонтальное.
- Тиражированная (англ. *replicated database*): методом распределения данных является тиражирование (репликация).

Другие виды БД

- Пространственная (англ. *spatial database*): БД, в которой поддерживаются пространственные свойства сущностей предметной области. Такие БД широко используются в геоинформационных системах.
- Временная, или темпоральная (англ. *temporal database*): БД, в которой поддерживается какой-либо *аспект времени*, не считая времени, определяемого пользователем.
- Пространственно-временная (англ. *spatial-temporal database*) БД: БД, в которой одновременно поддерживается одно или более измерений в аспектах как пространства, так и времени.
- Циклическая (англ. *round-robin database*): БД, объём хранимых данных которой не меняется со временем, поскольку в процессе сохранения новых данных они заменяют более старые данные. Одни и те же ячейки для данных используются циклически.

- Простейшая схема работы с базой данных:



- По характеру использования СУБД делят на однопользовательские (предназначенные для создания и использования БД на персональном компьютере) и многопользовательские (предназначенные для работы с единой БД нескольких компьютеров, объединенных в локальные сети). Вообще деление по характеру использования можно представить следующей схемой:



Структура базы данных

- Создавая базу данных, мы стремимся упорядочить информацию по различным признакам для того, чтобы потом извлекать из нее необходимые нам данные в любом сочетании. Сделать это возможно, только если данные структурированы. Структурирование - это набор соглашений о способах представления данных. Понятно, что структурировать информацию можно по-разному. В зависимости от структуры различают иерархическую, сетевую, реляционную, объектно-ориентированную и гибридную модели баз данных. Самой популярной на сегодняшний день является реляционная структура, поэтому об остальных упомянем лишь вскользь.

Иерархическая структура базы данных

- Это древовидная структура представления информации. Ее особенность в том, что каждый узел на более низком уровне имеет связь только с одним узлом на более высоком уровне. Посмотрим, например, на фрагмент иерархической структуры базы данных "Институт":



- Из структуры понятно, что на одной кафедре может работать несколько преподавателей. Такая связь называется "один ко многим" (одна кафедра - много преподавателей).

- Но если мы попытаемся добавить в эту структуру группы студентов, то нам понадобится связь *"многие ко многим"*:



- (один преподаватель может работать со многими группами, а одна группа может учиться у многих преподавателей), а такой связи в иерархической структуре быть не может (т.к. связь может быть только с одним узлом на более высоком уровне). Это основной недостаток подобной структуры базы данных.

Реляционная структура базы данных

Пользователи

id пользователя	Имя	E-mail	Пароль
1	Кирилл	kirill@mail.ru	*****
2	Вася	vasy@rambler.ru	*****
3	Семен	semen@yandex.ru	*****
4	Вася	vasy@mail.ru	*****

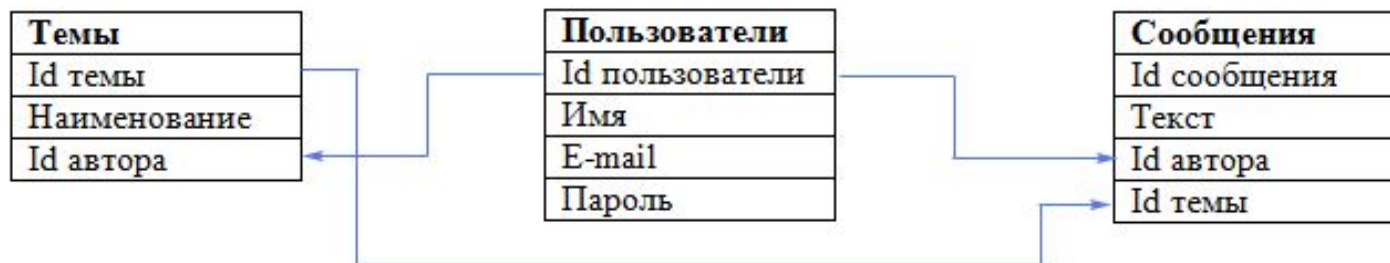
Все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные.

Темы

id темы	Наименование	id автора
1	О рыбалке	1
2	Велосипеды	2
3	Ночные клубы	3
4	О рыбалке	1
5	К кому обратиться	4

Сообщения

id сообщения	Текст	id автора	id темы
1	Думаю надо сделать так...	1	1
2	Согласен	2	4
3	А еще можно сделать так...	3	1
4	Согласен	2	1



Реляционные базы данных, как мы уже знаем, состоят из таблиц. Каждая таблица состоит из столбцов (их называют *полями* или *атрибутами*) и строк (их называют *записями* или *кортежами*). Таблицы в реляционных базах данных обладают рядом свойств. Основными являются следующие:

- В таблице не может быть двух одинаковых строк. В математике таблицы, обладающие таким свойством, называют *отношениями* - по-английски relation, отсюда и название - реляционные.
- Столбцы располагаются в определенном порядке, который создается при создании таблицы. В таблице может не быть ни одной строки, но обязательно должен быть хотя бы один столбец.
- У каждого столбца есть уникальное имя (в пределах таблицы), и все значения в одном столбце имеют один тип (число, текст, дата...).
- На пересечении каждого столбца и строки может находиться только атомарное значение (одно значение, не состоящее из группы значений). Таблицы, удовлетворяющие этому условию, называют *нормализованными*.

Объектно-ориентированные и гибридные базы данных

- В объектно-ориентированных базах данных данные хранятся в виде объектов, что очень удобно. Но на сегодняшний день такие БД еще распространены, т.к. уступают в производительности реляционным.

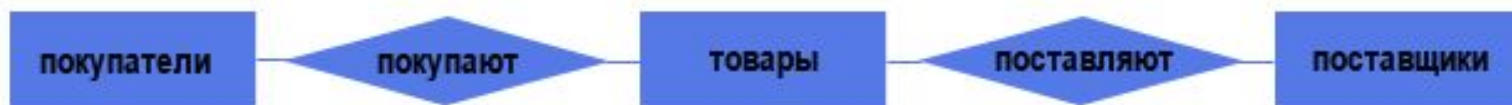
Гибридные БД совмещают в себе возможности реляционных и объектно-ориентированных, поэтому их часто называют объектно-реляционными.

Примером такой СУБД является Oracle, начиная с восьмой версии.

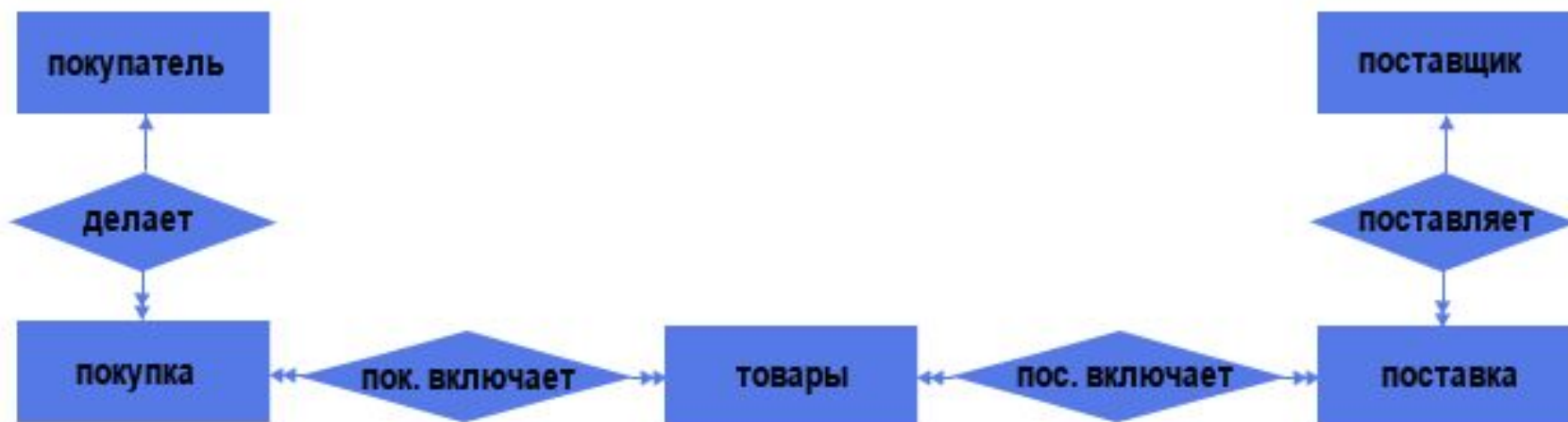
Концептуальная модель базы данных

- Концептуальная модель - это отражение предметной области, для которой разрабатывается база данных. Не вдаваясь в теорию, отметим, что это некая диаграмма с принятыми обозначениями элементов. Так, все объекты, обозначающие вещи, обозначаются в виде прямоугольника. Атрибуты, характеризующие объект - в виде овала, а связи между объектами - ромбами. Мощность связи обозначаются стрелками (в направлении, где мощность равна многим - двойная стрелка, а со стороны, где она равна единице - одинарная).

В качестве примера рассмотрим интернет-магазин. У магазина есть товары, которые поставляются поставщиками и покупаются покупатели. Это можно представить тремя объектами и двумя связями:

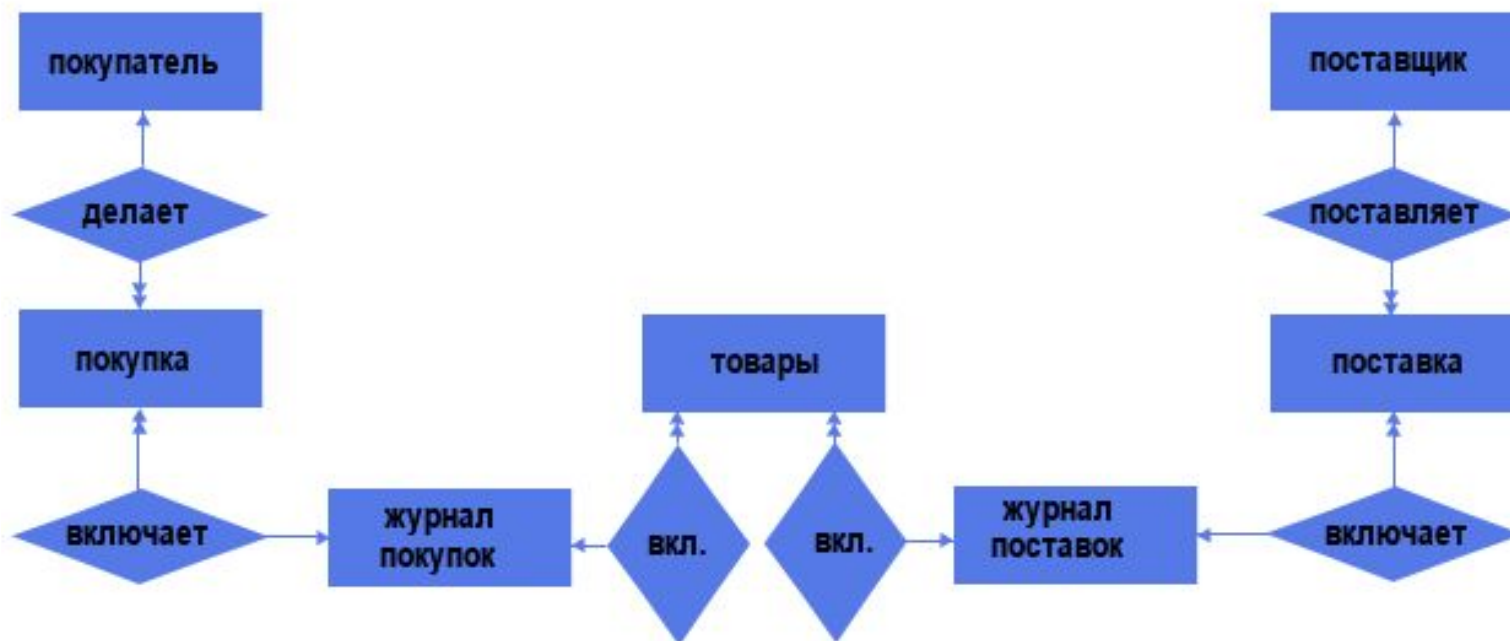


Но как поставщик поставляет товары? Он делает поставку, которая подтверждается документом. Аналогично и покупатель делает покупку, которая также может подтверждаться документом. Таким образом, поставка и покупка могут рассматриваться, как самостоятельные объекты:



Теперь у нас пять объектов и четыре связи. Две связи "один ко многим" (один поставщик может осуществить несколько поставок, но каждая поставка осуществляется только одним поставщиком, аналогично и для связи Покупатель - Покупка) и две связи "многие ко многим" (каждая поставка может содержать несколько товаров, а один и тот же товар может содержаться в нескольких поставках, аналогично и для связи Покупка - Товар).

Но связи "многие ко многим" недопустимы в реляционной модели, поэтому каждую такую связь надо заменить на две связи "один ко многим". Делается это добавлением промежуточного объекта:



Таким образом, у нас появилось еще два объекта - журнал покупок и журнал поставок, со связями "один ко многим" (один журнал поставок может включать несколько поставок, но каждая поставка может входить только в один журнал, аналогично и для остальных).

Каждый объект нашего магазина имеет свои атрибуты:



Заключение

Вообще, если предметная область обширная, то ее полезно разбить на несколько локальных предметных областей (наша концептуальная модель отражает именно локальную предметную область). Объем локальной области выбирается таким образом, чтобы в нее входило не более 6-7 объектов. После создания моделей каждой выделенной предметной области производится объединение локальных концептуальных моделей в одну общую, как правило, довольно сложную схему.