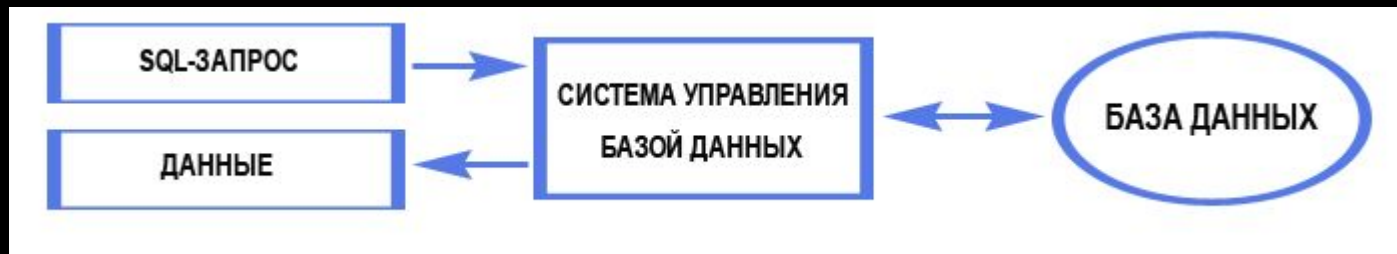


# БАЗЫ ДАННЫХ

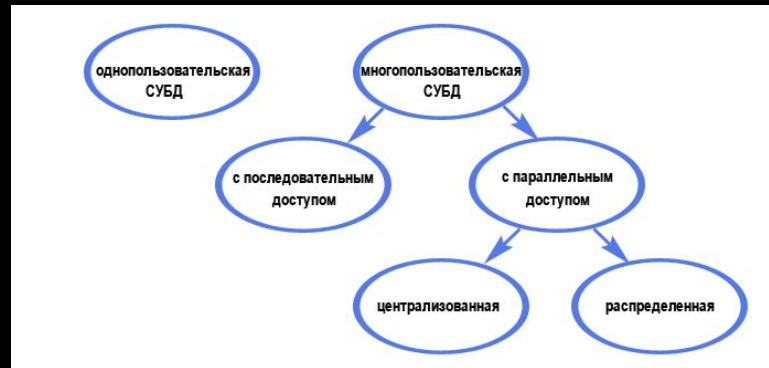
База данных - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных - это хранилище данных. Сами по себе базы данных не представляли бы интереса, если бы не было систем управления базами данных (СУБД).

- 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.
- Простейшая схема работы с базой данных выглядит примерно так:



- 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:



- 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.

- Не вдаваясь далее в подробности, отметим, что на сегодняшний день число используемых СУБД исчисляется десятками. Наиболее известные однопользовательские СУБД - Microsoft Visual FoxPro и Access, многопользовательские - MS SQL Server, Oracle и MySQL.

- 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.
- Сетевая структура базы данных - по сути, это расширение иерархической структуры. Все то же самое, но существует связь "многие ко многим". Сетевая структура базы данных позволяет нам добавить группы в наш пример. Недостатком сетевой модели является сложность разработки серьезных приложений

- Реляционная структура базы данных Все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные. Подробно об этом мы будем говорить в следующих уроках, здесь же хочется отметить, что эта структура стала настоящим прорывом в развитии баз данных.
- 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. Несомненно, такие БД будут развиваться в будущем, но пока первенство остается за реляционными структурами. Поэтому именно их мы и будем изучать в последующих уроках.



- 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.

- Первичный ключ (сокращенно РК - primary key) - столбец, значения которого во всех строках различны. Первичные ключи могут быть логическими (естественными) и суррогатными (искусственными). Так, для нашей таблицы Пользователи первичным ключом может стать столбец e-mail (ведь теоретически не может быть двух пользователей с одинаковым e-mail).

- На практике лучше использовать суррогатные ключи, т.к. их применение позволяет абстрагировать ключи от реальных данных. Кроме того, первичные ключи менять нельзя, а что если у пользователя сменится e-mail? Суррогатный ключ представляет собой дополнительное поле в базе данных. Как правило, это порядковый номер записи (хотя вы можете задавать их на свое усмотрение, контролируя, чтобы они были уникальны).

- 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).

- Если предметная область обширная, то ее полезно разбить на несколько локальных предметных областей. 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.

# ПРЕОБРАЗОВАНИЕ МОДЕЛИ В РЕЛЯЦИОННУЮ

- Преобразование концептуальной модели в реляционную состоит в следующем: Построить набор предварительных таблиц и указать первичные ключи. Провести процесс нормализации.
- Нормализация - это пошаговый, обратимый процесс замены исходной схемы другой схемой, в которой таблицы имеют более простую и логичную структуру.
- «Аномалии обновления» – при работе с таблицами могут возникнуть различного рода аномалии

Примеры:

«Аномалия удаления» - При удалении одного из сообщений таблицы вместе может удалиться и информация о теме.

«Аномалия модификации» - Если мы решим поменять название темы, то нам придется просмотреть все строки и в каждой заменить старую тему на новую

Существуют и многие другие аномалии

- Далеко не всегда эти недостатки можно учесть сразу. Для их устранения и применяется процесс нормализации. Он включает ряд правил, используемых для проверки всех таблиц базы данных.
- Различают:
- 1НФ - первая нормальная форма
- 2НФ - вторая нормальная форма
- 3НФ - третья нормальная форма
- НФБК - нормальная форма Бойса-Кодда
- 4НФ - четвертая нормальная форма
- 5НФ - пятая нормальная форма



- Каждая нормальная форма налагает определенные ограничения на данные. Каждая нормальная форма более высокого уровня предполагает, что анализируемая таблица уже находится в нормальной форме на уровень ниже рассматриваемой. В ходе нормализации схема базы данных становится все более строгой, а ее таблицы все менее подвержены различного рода аномалиям. Для реляционных баз данных необходимо, чтобы ее таблицы находились в 1НФ. Нормальные формы более высоких уровней могут использоваться разработчиками по своему усмотрению. Однако грамотный специалист стремится к тому, чтобы довести уровень нормализации базы данных хотя бы до 3НФ, тем самым исключив избыточность данных и аномалии обновления. Надо сказать, что НФБК, 4НФ и 5НФ используются крайне редко.

# ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

- Таблица находится в первой нормальной форме, если все ее поля имеют простые (атомарные) значения. Само понятие атомарности определить достаточно трудно. Значение, атомарное в одном случае, может быть неатомарным в другом. Общий принцип здесь такой: значение не атомарно, если оно используется по частям.

# ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

- Эта форма применяется к таблицам с составными ключами. Таблица, у которой первичный ключ включает только одно поле, всегда находится во 2НФ. Таблица находится во второй нормальной форме, если она находится в первой нормальной форме, а каждое неключевое поле функционально полно зависит от составного ключа.

# ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

- Таблица находится в третьей нормальной форме, если она находится во второй нормальной форме, и каждое неключевое поле нетранзитивно зависит от первичного ключа. Транзитивная зависимость наблюдается в том случае, если одно из двух неключевых полей зависит от первичного ключа, а другое зависит от первого неключевого поля.

- Подготовил презентацию студент группы МОИС-251 Сопыряев Сергей