**Базы данных и их разновидности**

**База данных (БД)**–это совокупность массивов и файлов данных, организованная по определённым правилам, предусматривающим стандартные принципы описания, хранения и обработки данных независимо от их вида.

**Основные классификации баз данных**

Существует огромное количество разновидностей баз данных, отличающихся по различным критериям. Основные из них:

1. **Классификация по модели данных**

Центральным понятием в области баз данных является понятие *модели*.

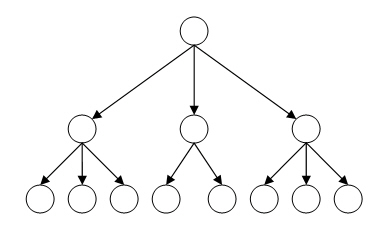
**Модель данных**- это некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

**Виды:**

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

**1) Иерархическая база данных** – каждый объект, при таком хранение информации, представляется в виде определенной сущности, то есть у этой сущности могут быть дочерние элементы, родительские элементы, а у тех дочерних могут быть еще дочерние элементы, но есть один объект, с которого все начинается. Получается своеобразное дерево. Примером иерархической базы данных может быть документ в формате XML или файловая система компьютера.

Следует сказать, что базы данных подобного вида оптимизированы под чтение информации, то есть базы данных, имеющие иерархическую структуру умеют очень быстро выбирать запрашиваемую информацию и отдавать ее пользователям. Но такая структура не позволяет столь же быстро перебирать информацию. Здесь можно привести первый пример из жизни: компьютер может легко работать с каким-либо конкретным файлом или папкой (которые, по сути, являются объектами иерархической структуры), но проверка компьютера антивирусам осуществляется очень долго. Второй пример – реестр Windows.



На изображении Вы можете увидеть структуру иерархической базы данных. В самом верху находится *родитель* или *корневой элемент*, ниже находятся дочерние элементы, элементы находящиеся на одном уровне называются *братьями* или *соседними элементами*. Соответственно, чем ниже уровень элемента, тем вложенность этого элемента больше.

**Объектные базы данных** - это модель работы с объектными данными.

Такая модель баз данных, несмотря на то, что она существует уже много лет, считается новой. И её создание открывает большие перспективы, в связи с тем, что использование объектной модели баз данных легко воспринимается пользователем, так как создается высокий уровень абстракции. Объектная модель идеально подходит для трактовки такого рода объектных данных как изображение, музыка, видео, разного вида текст.

**Объектно-ориентированная база данных (ООБД)**- база данных, в которой данные моделируются в виде объектов, их атрибутов, методов и классов.

Объектно-ориентированные базы данных обычно рекомендованы для тех случаев, когда требуется высокопроизводительная обработка данных, имеющих сложную структуру.

**2) Объектно-реляционные СУБД** объединяют в себе черты реляционной и объектной моделей. Их возникновение объясняется тем, что реляционные базы данных хорошо работают со встроенными типами данных и гораздо хуже - с пользовательскими, нестандартными. Когда появляется новый важный тип данных, приходится либо включать его поддержку в СУБД, либо заставлять программиста самостоятельно управлять данными в приложении.

Не всякую информацию имеет смысл интерпретировать в виде цепочек символов или цифр. Представим себе музыкальную базу данных. Песню, закодированную в виде аудиофайла, можно поместить в текстовое поле большого размера, но как в таком случае будет ли осуществляться текстовый поиск?

**3) Реляционная(или табличная) БД** содержит перечень объектов одного типа, т.е. объектов с одинаковым набором свойств.

Такую базу удобно представлять в виде двумерной таблицы (или, чаще всего, нескольких связанных между собой таблиц).

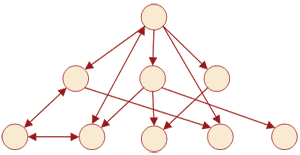
Примером такой таблицы может служить БД "Учащиеся", представляющая собой перечень объектов (учеников), каждый из которых имеет фамилию, имя, отчество, дату рождения, класс, номер личного дела и др.

Столбцы такой таблицы называют полями; каждое поле характеризуется своим именем (названием соответствующего свойства объекта) и типом данных, которые это поле может хранить. Каждое поле обладает определенным набором свойств (размер, формат и т. п.). Т. о., *поле БД* - это столбец таблицы, содержащий значения определенного свойства объектов.

Строки таблицы являются записями. Записи разбиты на поля. Каждая строка таблицы содержит запись об одном единственном объекте, включая все его свойства.

В каждой таблице должно быть хотя бы одно ключевое поле, содержимое которого уникально для любой записи в этой таблице. Значения ключевого поля однозначно определяют каждую запись в таблице. В приведенном выше примере ключевым полем может являться поле "Номер личного дела". Очень часто в качестве ключевого поля используется поле, содержащее данные типа счетчик.

**4) Сетевые базы данных**являются своеобразной модификацией иерархических баз данных. Если Вы внимательно смотрели на изображение выше, то наверняка обратили внимание, что к каждому нижнему элементу идет только одна стрелочка от верхнего элемента. То есть у иерархических баз данных у каждого дочернего элемента может быть только один потомок. Сетевые базы данных отличаются от иерархических тем, что у дочернего элемента может быть несколько предков, то есть элементов стоящих выше него. Для большей наглядности и понимания структуры сетевых баз данных обратите внимание на изображение:



Стоит заметить, что сетевые базы данных обладают примерно теми же характеристиками, что и иерархические базы данных. Но сейчас нас не особо интересуют иерархические и сетевые базы данных, данная тема больше относится к формату XML.

**5) Функциональные базы** данных используются для решения аналитических задач: финансовое моделирование и управление производительностью. Функциональная база данных или функциональная модель отличается от реляционной модели. Функциональная модель также отличается от других аналогично названных концепций, включая модель функциональной базы данных DAPLEX и базы данных функциональных языков.

Функциональная модель является частью категории оперативной аналитической обработки (OLAP электронной таблице,), поскольку она включает многомерное иерархическое объединение. Но она выходит за рамки OLAP, требуя ориентирования ячейки, подобно тому, где ячейки могут быть введены или рассчитаны как функции других ячеек. Также, как и в электронных таблицах, данная модель поддерживает интерактивные вычисления, в которых значения всех зависимых ячеек автоматически обновляются каждый раз, когда изменяется значение ячейки.

1. **Классификация по содержимому**

Примеры:

* Географическая.
* Историческая.
* Научная.
* Мультимедийная.
* Клиентская.
  1. **Классификация по степени распределённости:**
* Централизованная или сосредоточенная (англ. centralized database): БД, которая полностью поддерживается на одном компьютере.
* Распределённая (англ. distributed database): БД, 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.
  1. **Классификация БД по среде физического хранения:**
* БД во вторичной памяти (традиционные): средой постоянного хранения является периферийная энергонезависимая память (вторичная память) - это, как правило, жёсткий диск. В оперативную память СУБД помещает лишь кэш и данные для текущей обработки.
* БД в оперативной памяти (in-memory databases): все данные находятся в оперативной памяти.
* БД в третичной памяти (tertiary databases): средой постоянного хранения является отсоединяемое от сервера устройство массового хранения (третичная память), как правило, на основе магнитных лент или оптических дисков. Во вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кэш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры.

**SQL**

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

**Функции языка SQL:**

* Организация данных – создание и изменение структуры баз данных.
* Чтение данных.
* Обработка данных – удаление, добавление и корректировка данных.
* Управление доступа к данным – предоставление привилегий (ограничение возможностей) пользователю для чтения и изменения данных.
* Совместное использование данных - координация общего пользования данных многими пользователями.
* Целостность данных – защита данных от разрушения при сбое системы или других обстоятельствах.

**СУБД**

Большинство современных СУБД построено на реляционной модели данных. Для получения информации из отношений (таблиц) базы данных в качестве языка манипулирования данными в теоретическом плане используется язык SQL

**СУБД** - система управления базами данных, совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных

**Основные функции СУБД:**

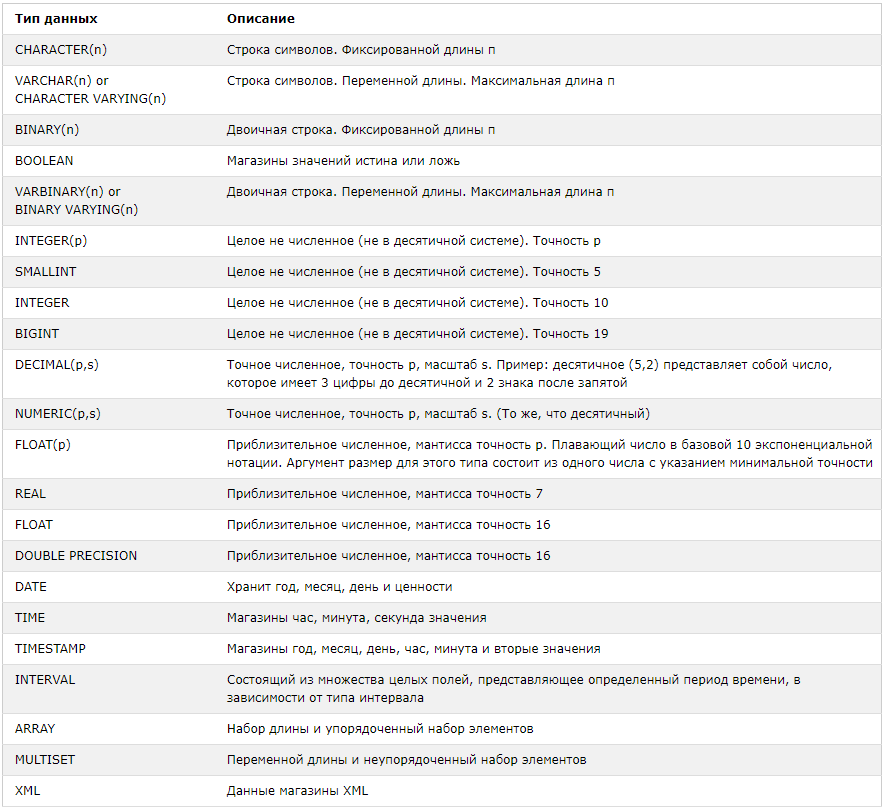
* Управление данными во внешней памяти (на дисках).
* Управление данными в оперативной памяти с использованием дискового кэша.
* Журнализация изменений, резервное копирование и восстановление базы данных после сбоев.
* Поддержка языков БД (язык определения данных, язык манипулирования данными).

**Типы данных в SQL**

Каждый столбец в таблице базы данных должен иметь имя и тип данных.

SQL разработчики должны решить, какие типы данных будут храниться внутри каждого столбца таблицы при создании таблицы SQL. Тип данных представляет собой метку и ориентир для SQL, чтобы понять, какой тип данных, как ожидается, внутри каждого столбца, а также определяет, как SQL будут взаимодействовать с хранимыми данными.

В следующей таблице перечислены общие типы данных в SQL:

****

**SQL Data Type - Краткий справочник в разрезе БД**

Тем не менее, различные базы данных предлагают различные варианты для определения типа данных.

В следующей таблице приведены некоторые из общих названий типов данных между различными платформами баз данных:

