1. База данных, определение, основные понятия. Типы организации базы данных.

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

**Понятия:**

Таблицы – главные объекты базы данных. Они состоят из строк и столбцов и содержат основные данные, хранящиеся в БД.

Запросы — это один из объектов баз данных, который служит для манипулирования данными в таблице, к ним относят добавление, удаление и изменение записей.

Поле – это наипростейший элемент таблицы базы данных, в нем храниться значение объекта или процесса.

Сущность – это некий объект, информацию о котором можно узнать из базы данных.

Атрибут – свойство сущности

Связь — это объединение, которое показывает между некоторыми сущностями как они взаимодействуют друг с другом.

**Типы:**

Иерархический тип, ее особенность в том, что каждая запись имеет только ОДНОГО «родителя».

Сетевой тип имеет расширенный функционал иерархической модели и в них элементы имеют более одной связи.

Один из самых распространенных типов – это реляционная модель, она представлена в виде таблиц, которые имеют атрибуты.

Объектно-ориентированный –  в которой данные моделируются в виде объектов, их атрибутов, методов и классов.

1. Типы и виды запросов пользователей.

**Запрос на выборку** – это самый простой и потому наиболее часто используемый вид запроса. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновить.

**Запрос с параметрами** – это запрос, отображающий при выполнении в собственном диалоговом окне приглашение ввести данные (например, значение, которое требуется вставить в поле).

**Запрос на изменения** – это запрос, который за одну операцию изменяет или перемещает несколько записей. К ним относят – запросы на удаление, обновление, добавление записей, запросы на создание таблиц.

**Запрос с критерием поиска** — позволяет производить отбор записей в соответствии с заданным критерием поиска

1. Администратор БД, основные функции, состав группы, определение.

Администратор базы данных — лицо, отвечающее за выработку требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение, включая управление учётными записями пользователей БД и защиту от несанкционированного доступа.

**в основные задачи АБД входит:**

* Проектирование базы данных.
* Оптимизация производительности базы данных
* Обеспечение и контроль доступа к базе данных
* Обеспечение безопасности в базе данных
* Резервирование и восстановление базы данных
* Обеспечение целостности баз данных
* Обеспечение перехода на новую версию СУБД

**Типы администраторов:**

* Системный администратор
* Архитектор БД
* Аналитик БД
* Разработчик моделей данных
* Администратор приложении
* Аналитик производительности
* Администратор хранилища данных

1. Режимы работы с БД.

Все работы с базой данных можно разделить на два основных режима их проведения. Это эксплуатационный и проектировочный режимы.

Проектировщик БД может создавать в ней новые объекты (например, формы, макросы или таблицы), задавать их структуру, изменять свойства существующих полей или добавлять новые, устанавливать дополнительные связи между таблицами.

Вторая категория исполнителей, работающих с базами данных, — *пользователи*. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы — только к данным, да и то не ко всем, а к тем, работа с которыми предусмотрена на конкретном рабочем месте.

1. Понятие [банка данных](https://pandia.ru/text/category/bank_dannih/). Основные компоненты банка данных. Требования к банку данных со стороны пользователей.

Банк данных (БнД) - это система специально организованных данных, программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного многоцелевого использования данных.

**Банк данных в общем случае состоит из следующих компонентов:**

* БД - представляет собой совокупность специальным образом организованных данных хранимых в памяти ВС(вычислительной системы) и отображающих состояние объектов и их взаимосвязи в рассматриваемой предметной области.
* СУБД – это комплекс языковых и программных средств, предназначенных для создания, ведения и совместного использования базы данных многими пользователями.
* Приложения – представляют собой программу или комплекс программ, обеспечивающий автоматизацию обработки информации для прикладной задачи. Их разрабатывают в случаях, когда требуется обеспечить удобство работы с БД неквалифицированным пользователям или интерфейс СУБД не устраивает пользователя.
* Словарь данных – представляет собой подсистему банка данных, предназначенную для централизованного хранения информации о структурах данных, взаимосвязях файлов друг с другом, типах данных, форматах их представления, принадлежности данных пользователям.
* Администратор БД – есть лицо или группа лиц, отвечающее за выбор требований к БД, ее проектирование, создание, эффективное использование и сопровождение.
* Вычислительная система– представляет собой совокупность взаимосвязанных и согласованно действующих компьютеров и других устройств, обеспечивающих автоматизацию процессов приема, обработки и выдачи информации потребителю.
* Обслуживающих персонал – выполняет функции поддержания работы технических и программных средств, работоспособность их состояния

**Требования к банку данных:**

* Обеспечивать возможность хранения и модификации больших объемов информации. Удовлетворять сегодняшним и вновь возникающим требованиям со стороны пользователя.
* Обеспечивать заданные уровни достоверности и непротиворечивости хранимой информации.
* Обеспечивать доступ к данным только тех пользователей, которые имеют соответствующие полномочия.
* Обеспечивать возможность поиска информации.
* Удовлетворять заданным требованиям производительности при обработке запросов.
* Иметь возможность реорганизации и расширения при изменении границ предметной области.
* Обеспечивать возможность одновременного обслуживания большого числа внешних пользователей.

1. Трехуровневая архитектура описания базы данных. Режимы работы с базой данных.

В настоящее время чаще всего поддерживается трехуровневая архитектура описания БД, с тремя уровнями абстракции, на которых можно рассматривать базу данных. Такая архитектура включает:

* внешний уровень(пользовательский), на котором пользователи воспринимают данные, где отдельные группы пользователей имеют свое представление (ПП) на базу данных. Отделение внешнего уровня от концептуального обеспечивает *логическую независимость* данных;
* внутренний уровень(физический), на котором СУБД и операционная система воспринимают данные, а также позволяет скрыть подробности физического хранения данных (носители, файлы, таблицы, триггеры …) от концептуального уровня;
* концептуальный уровень(промежуточный) представления данных, предназначенный для отображения внешнего уровня на внутренний уровень, а также для обеспечения необходимой их независимости друг от друга; он связан с обобщенным представлением пользователей. Отделение внутреннего уровня от концептуального обеспечивает так называемую физическую независимость данных.

Режим работы с бд: см. вопрос №4.

1. Модели данных: понятие, основные компоненты и классификация.

Модель данных – совокупность структур данных и операций по их обработке.

**Любая модель данных должна содержать 3 компонента:**

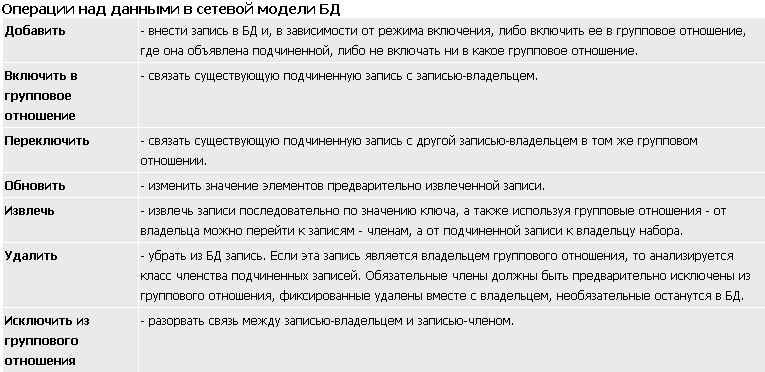
* Структура данных – представление хранимых данных, используемых в СУБД
* Набор допустимых операций, выполняемых по структуре данных
* Ограничения целостности – складываются из 2 понятий: целостности сущности и ссылочной целостности: **Целостность сущности** – каждый экземпляр сущности объекта реального мира различим и распознаваем: идентификатор объектов задаётся первичным ключом таблиц (сущностей). **Ссылочная целостность** (ссылочное ограничение) – ограничение, согласно которому значения внешних ключей связанных таблиц должны быть адекватны значениям первичного ключа главной таблицы связей

**Модели данных бывают:**

* **Сетевая модель** представляет данные в виде произвольного графа записи.
* **Иерархическая модель** – данные представлены в виде иерархической структуры.
* **Реляционная модель** (возникла от relation - отношение) – двумерная таблица, используемая для хранения данных об объектах реального мира.
* Объектно-ориентированные моделисочетают сетевые и реляционные модели.

8.  Сетевая модель данных: элементы структуры, основные операции над данными и ограничения целостности.

Сетевая модель данных — логическая модель данных, являющаяся расширением иерархического подхода Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков.

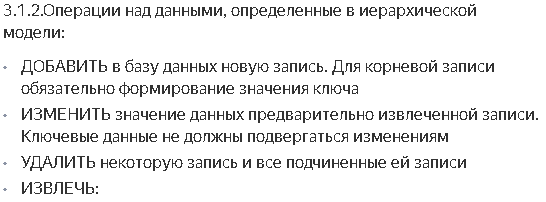


Ограничение целостности идентично иерархической модели.

 9.  Иерархическая модель данных: элементы структуры, основные операции над данными и ограничения целостности.

Иерархическая модель данных — это модель данных, где используется представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

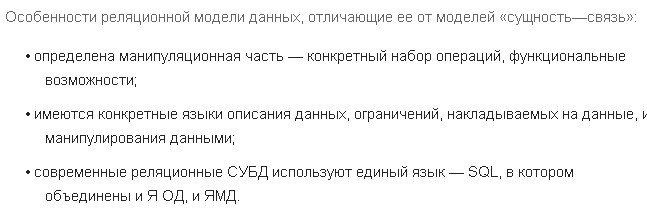
Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня)



**Поддерживается только целостность** связей между владельцами и членами группового отношения (никакой потомок не может существовать без предка). Не обеспечивается автоматическое поддержание соответствия парных записей, входящих в разные иерархии.

10.  Реляционная модель данных. Особенности реляционной модели. Правила Кодда.

Реляционная модель - совокупность данных, состоящая из набора двумерных таблиц.



**12(13) правил Кодда**

1. Реляционная СУБД должна быть способна полностью управлять базой данных, используя связи между данными.

2. Информационное правило - Вся информация в реляционной БД (включая имена таблиц и столбцов) должна определяться строго как значения таблиц.

3. Гарантированный доступ - Любое значение БД должно быть гарантированно доступным через комбинацию имени таблицы, первичный ключ и имя столбца.

4. Поддержка нулевого значения - СУБД должна уметь работать с нулевыми (пустыми) значениями. Нулевое значение - это неизвестное, независимое, неприменимое значение, в отличие от значений по умолчанию и обычных значений.

5. Активный, оперативный реляционный каталог - Описание БД и его содержимое должны быть определены на логическом уровне через таблицы, к которым можно применять запросы, используя DML (язык манипулирования данными).

6. Исчерпывающее подмножество языка данных - По крайней мере, один из поддерживаемых языков должен иметь четко определенный синтаксис и быть самодостаточным. Он должен поддерживать определение данных и манипулирование ими, правила целостности, авторизацию и транзакции.

7. Правило обновления представлений - Все представления, теоретически обновляемые, могут быть обновлены через систему.

8. Вставка, обновление и удаление - СУБД поддерживает не только запрос данных, но и вставку, обновление и удаление.

9. Физическая независимость данных - Логика программ-приложений остается прежней при изменении физических методов доступа к данным и структур хранения.

10. Логическая независимость данных - Логика программ-приложений остается прежней, в пределах разумного, при изменении структур таблиц.

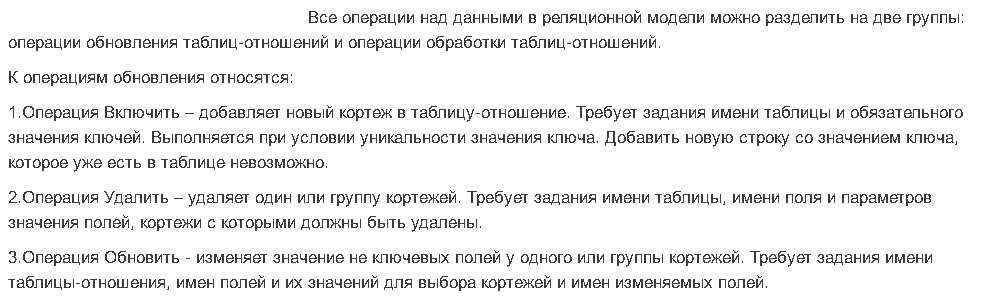
11. Независимость целостности - Язык БД должен быть способен определять ограничения целостности. Они должны быть доступны из оперативного каталога, и не должно быть способа их обойти.

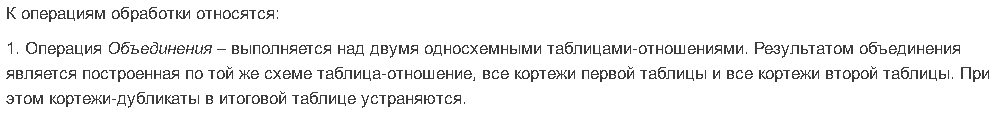
12. Независимость распределения - Запросы программ-приложений логически не затрагиваются при первом и последующих распределениях данных.

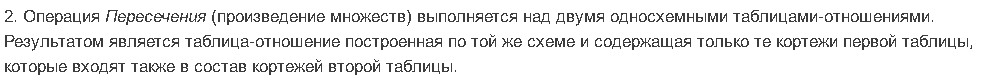
13. Несмешиваемость (может, кто-то знает более удачный перевод термина ?) - невозможность обойти ограничения целостности, используя языки низкого уровня.

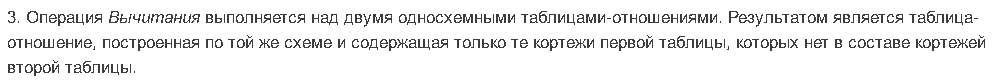
11.  Реляционная модель данных: элементы структуры, основные операции над данными и ограничения целостности. Виды отношений и ключей.

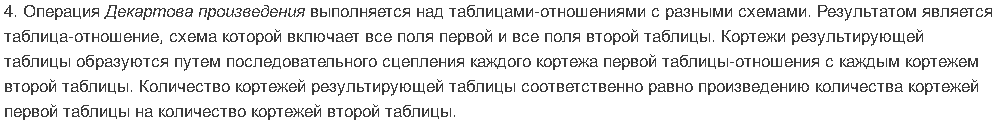
Реляционная таблица является основным типом структуры данных (объектом) реляционной модели. Структура этой таблицы определяется совокупностью столбцов (полей), для которых определен тип данного. Основной логической единицей обработки (поиск, выборка, сортировка, вычисления) в реляционной БД является строка таблицы (запись).











**Перви́чный** **ключ** (англ. **primary** **key**) — **в** реляционной модели **данных** один из потенциальных **ключей** отношения, выбранный в качестве основного **ключа**

**Вторичный** **ключ** - **это** одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле или поля первичного **ключа** в другой таблице.

**Свзяи:**

Связь многие-ко-многим – это связь, при которой множественным записям из одной таблицы (A) могут соответствовать множественные записи из другой (B)

В связи один-к-одному каждый блок сущности A может быть ассоциирован с только 1 блоком сущности B.

Связь один ко многим. Это наиболее часто встречаемый тип связей. В этом типе связей несколько строк из дочерний таблицы зависят от одной строки в родительской таблице.

12.  Основы реляционной алгебры. Операции над отношениями.

13.  Основные принципы проектирования баз данных. Этапы проектирования: инфологическое моделирование, даталогическое проектирование, физическое проектирование.

14.  Системный анализ предметной области. Информационно-логическое проектирование.

15.  Избыточность данных и аномалии обновления в базе данных. Функциональные зависимости между атрибутами.

16.  Нормализация отношений. Преобразование ER-модели в схему реляционной базы данных.

17.  Физическое проектирование. Особенности, влияющие на организацию внешней памяти. Технологии хранения данных.

18.  СУБД: основные функции, типы. Свойства и сравнительные характеристики СУБД.

19.  Модели «клиент-сервер» в технологии БД. Схема, основные функции клиента, понятие сервера и клиента.

20.  Модель файлового сервера. Схема, основные функции клиента, сервера. Способ организации обмена данными между клиентом и сервером, преимущества, недостатки.

21.  Модель сервера баз данных. Схема, основные функции клиента, сервера. Способ организации обмена данными между клиентом и сервером, преимущества, недостатки.

22.  Автоматизированные системы проектирования баз данных. Основные возможности Case-средств. Классификация Case-средств.

23.  Обеспечение функционирования БД. Транзакции: понятия, модели завершения, свойства. Управление транзакциями.

24.  Обеспечение функционирования БД. Журнализация: откат транзакции, восстановление данных в результате сбоев.

25.  Обеспечение функционирования БД. Проблемы многопользовательских систем. Конфликты между транзакциями.

26.  Обеспечение функционирования БД. Триггеры: понятие, правила создания. Хранимые процедуры: понятие, виды, преимущества использования.

27.  Администрирование БД.

28.  Хранилище данных.

29.  Объектно-реляционная модель данных. Объектно-ориентированная модель данных.

30.  Отличие SQL от процедурных языков программирования. Интерактивный и встроенный SQL. Составные части SQL.