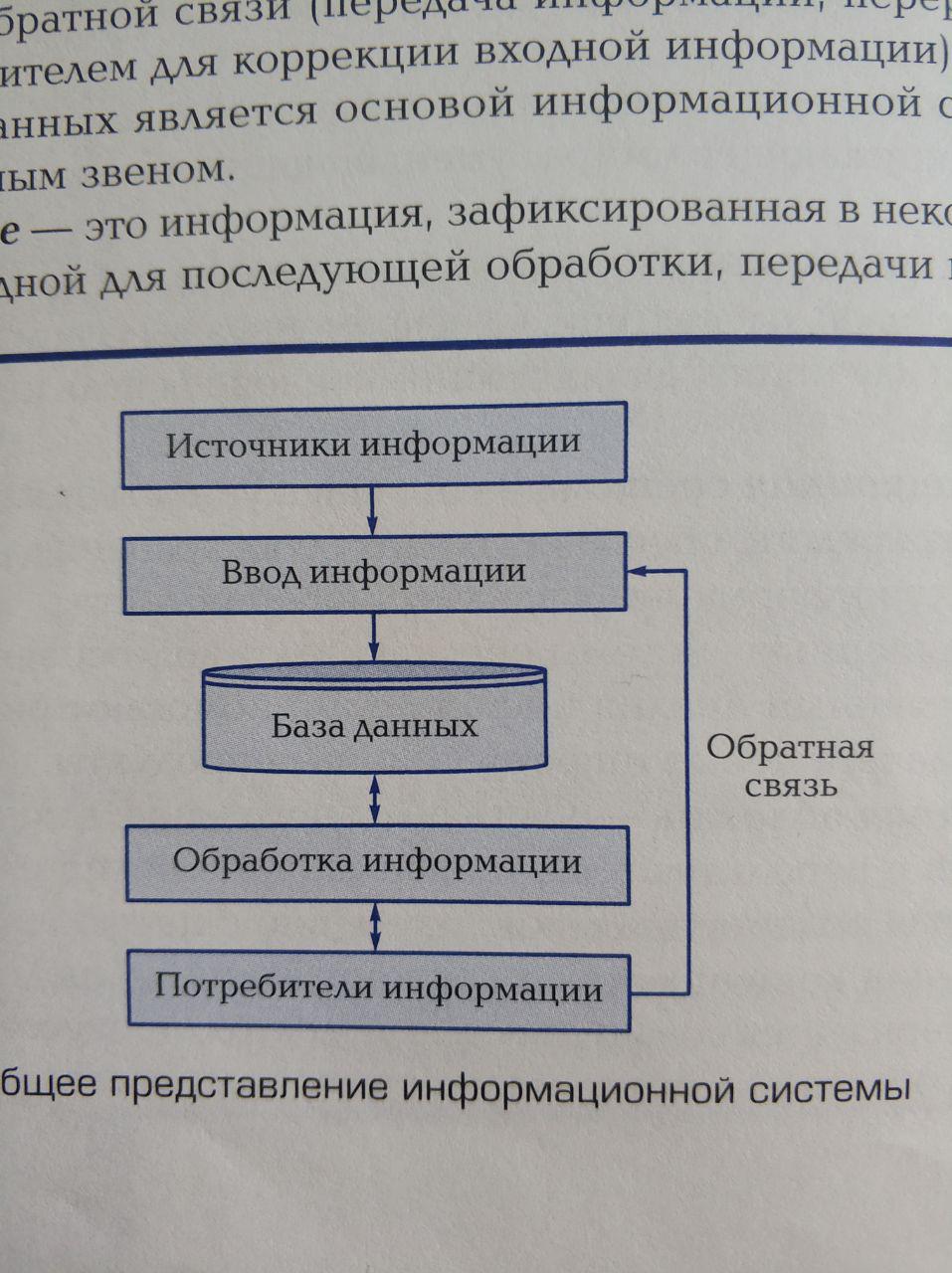
1. Базы данных и информационные системы. Основные определения.

Информация - это сведения об объектах, явлениях, процессах, событиях окружающего мира уменьшающие степень неопределенности знаний о них. Своевременная выдача достоверной информации для принятия решений это основная цель информационных систем. Информационная система - это совокупность технических и программных средств, обеспечивающих сбор, хранение, обработку, поиск, выдачу информации в задачах любой области. Предметная область - часть реального мира, данные о которой хранятся и используются в информационной системе. Предметная область подлежит изучению с целью организации управления и автоматизации. Она характеризуется совокупностью объектов, использующих их процессов и множеством пользователей. Ее анализ предшествует созданию любой информационной системы.

Информационный объект - это описание некоторой сущности предметной области.

Информационный объект можно представить в виде схемы.



Данные - это информация, зафиксированная в некоторой форме, пригодная для дальнейшей машиной обработки, передачи и хранения. База данных - это именованная совокупность взаимосвязанных данных отображающая состояние объектов в некоторой предметной области. Структурой данных называют совокупность правил и ограничений, которые отображают связи между отдельными частями данных. Обработка данных - совокупность задач для преобразования массивов данных. Система обработки данных - набор аппаратных и программных средств, осуществляющих выполнение задач по управлению данными. Управление данными - это круг операций с ними, которые необходимы для успешного функционирования системы обработки данных. Метаданные - это данные о данных, это описание собственной структуры базы данных.

2. Системы управления базами данных.

Это совокупность языковых и программных средств, предназначенных для управления, создания и использования баз данных. Они являются инструментальными средствами для извлечения данных, их обработки, изменения и анализа. СУБД предоставляет удобный, быстрый и контролируемый доступ к данным, обладает средствами для обеспечения целостности данных, поддержания баз данных в рабочих состояниях, обеспечения секретности, восстановления и сохранности информации в многопользовательском окружении.

СУБД предназначена в основном для профессиональных разработчиков. Обычный пользователь работает с базами данных с помощью специальных приложений, которые могут создаваться как внутри СУБД, так и вне ее с помощью систем программирования. СУБД должна обеспечить параллельную независимую работу таких приложений с единой базой данных.

Более современной формой организации хранения и доступа к информации является банк данных. Банк данных - система специально организованных данных, включая базы данных, программных, языковых, организационно-методических средств для обеспечения централизованного накопления и коллективного многоцелевого использования.

Словарь данных - это подсистема банка данных, предназначенная для централизованного хранения информации о структурах данных, взаимосвязях файлы базы данных друг с другом и т.п.

3. Распределенные базы данных. Базы знаний.

Существуют базы данных, в которых предполагается хранение данных распределенное в компьютерной сети. Распределенная база данных – это база данных, включающая фрагменты из нескольких баз данных, которые располагаются на различных узлах компьютерной сети и возможно управляются различными СУБД. С точки зрения пользователей и прикладных программистов она выглядит как обычная локальная база данных. Распределенная структура повышает возможности совместного использования удаленных данных, надежность, доступность и производительность системы, улучшить ее масштабируемость.

Современные информационные системы зачастую используют базы знаний. Знания – выявленные закономерности предметной области, позволяющие решать задачи этой области. База знаний – совокупность знаний предметной области. Системы, использующие базы знаний называются интеллектуальными.

4. Основные категории пользователей и разработчиков баз данных.

Основная категория пользователей, для которых создается база данных – это конечные пользователи. Ими могут быть случайные одноразовые пользователи, регулярные пользователи. От конечных пользователей не должно требоваться специальных знаний в области вычислительной техники и программирования.

Разработчики и администраторы приложений. Эта группа пользователей функционирует во время проектирования, создания и реорганизации базы данных. Администраторы приложений координируют работу разработчиков. Разработчики конкретных приложений работают с той частью данных из базы, которая требуется для этих приложений. Наиболее сложные обязанности возложены на группу администратора баз данных – это группа пользователей, которые на начальной стадии разработки базы данных отвечает за ее оптимальную организацию и одновременную работу конечных пользователей. На стадии развития и реорганизации они отвечают за возможность корректной реорганизации базы данных без нарушения текущей ее эксплуатации.

В составе группы администратора базы данных должны быть:

1. Системные аналитики.

2. Проектировщики структур данных и внешнего информационного обеспечения.

3. Проектировщики технологических процессов обработки данных.

4. Системные и прикладные программисты.

5. Операторы и специалисты по техническому обслуживанию.

5. Основные функции СУБД.

Изначально системы управления базами данных пытались построить по принципам файловых систем, но получившаяся СУБД оказалась громоздкой и неэффективной. Поэтому были разработаны системы обработки данных. Они включили в себя сами данные, СУБД и прикладное ПО, которое обращается к данным через СУБД. Общими функциями СУБД являются:

1. Управление данными во внешней памяти.

2 Управление данными в ОП.

3. Управление транзакциями.

4. Журнализация, резервное копирование и восстановление.

5. Поддержка языков баз данных.

6. Управление данными во внешней памяти.

СУБД должна представлять пользователям возможность:

Сохранять извлекать и обновлять данные в базу данных - это самая фундаментальная функция назначения СУБД. Способ реализации этой функции должен быть скрыт от конечного пользователя.

Контролировать доступ к данным – это возможность обеспечить только санкционированной доступ к базе данных, используя поддержку уровня доступа к БД и отдельным её элементам. Каждый пользователь должен иметь доступ только к тем данным, которые доступны для него в соответствии с его пользовательскими правами.

Обеспечивать параллельную работу нескольких пользователей – СУБД имеют механизмы, которые гарантируют корректное выполнение данных многими пользователями при одновременном доступе.

Поддерживать целостность данных – осуществляется инструментальными средствами контроля для того чтоб данные и их изменения соответствии заданным правилам. Целостность базы данных это её свойство, означающее, что в ней задерживается полное не противоречие и адекватное отражающее требуемую область информации. Целостное состояние БД описывается с помощью ограничений целостности в виде условий, которым должны удовлетворять данные.

7. Управление данными в ОЗУ.

Обычно размер базы данных значительно больше объема оперативной памяти. Если при обращении к любым данным будет производиться обмен с внешней памятью, то скорость работы СУБД будет равно скорости работы внешней памяти. Единственным реальным способом увеличения этой скорости является буферизация в оперативную память.

Буфер - область оперативной памяти, в которых временно хранятся фрагменты базы данных, данные из которых предлагается использовать, при обращении к СУБД или записать в базу после обработки.

8. Управление транзакциями.

Транзакции это число действий над базами данных, которые последовательно выполняются по принципу все или не одного. Если транзакции выполняются успешно, то СУБД фиксирует произведенные действия БД во внешней памяти. Если хотя бы одно действие транзакций не выполняется, то все изменения произведенные ей в БД отменяются. Это необходимо для поддержания логической целостности БД.

При параллельном выполнении транзакций возможно возникновение конфликтов, разрешение которых является функцией СУБД, при обнаружении таких случаев обычно производится отмена изменений, произведенных одной или нескольких транзакций - **откат транзакций**.

9. Журнализация, резервное копирование и восстановление.

Одним из основных требований к СУБД является надёжность хранения данных во внешней памяти, в том числе защита их физической и логической целостности. Это значит, что СУБД должна представлять возможность восстановить последнее согласованное состояние БД после любого сбоя. Защита физической целостности включает в себя журнализацию изменений, резервное копирование и восстановление БД. **Журнализация изменений** - это последовательная запись во внешнюю память всех изменений, выполняемых в БД: порядковый номер, тип и время изменения; идентификатор транзакций, объект, подвергшийся изменению, предыдущее и новое состояние блокнота.

Журнал содержит отметки начала и завершения транзакций и отметки принятия контрольной точки, он не доступен пользователями СУБД. Для повышения надёжности могут поддерживаться 2 копии журнала.

**Резервное копирования базы данных** – это процесс копирования базы данных на носителе, предназначенное для восстановления данных, в оригинальном или в новом месте их расположения, в случае повреждения или разрушения.

**Восстановление базы данных** – это функция СУБД, которая после логических или физических сбоев приводит базу данных в актуальное состояние.

10. Поддержка языков баз данных.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий средства для работы с БД и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком реляционной СУБД в настоящее время является SQL. Кроме своих основных функций СУБД предоставляет вспомогательные утилиты для эффективного администрирования базы данных:

1. Отвечает за экспорт и импорт данных.

2. **Мониторинг базы данных** – отслеживание характеристик, функционирования и использование БД.

3. **Статистический анализ** – для оценки производительности или степень использования БД.

4. Реорганизация индекса

5. Сборка не используемых записей и перераспределение памяти для физического устранения удаленных записей с внешних носителей, дефрагментация освобождённого пространства.

В современных СУБД можно выделить ядро СУБД, процессор языка, БД, подсистему поддержки выполнения, внешние утилиты для выполнения возможности БД.

Ядро СУБД это набор программных модулей необходимый и достаточный для создания и поддержания БД. Сервисные программы представляют дополнительные услуги, зависящие от конкретной предметной области и потребности конкретного пользователя. Процессор языка БД компилирует операторы языка в программный машинный код.

11. Архитектура баз данных.

В 1975 году на заседании Американского Национального Института стандартов была предложена обобщенная трехуровневая модель архитектуры СУБД. Она включала внешний, концептуальный и физический уровни. Основная цель этой идеи заключается в отделении пользовательского представления базы данных от ее физического представления.

**Уровень внешних моделей** – это самый верхний уровень, где каждая модель имеет свое представление данных. Отдельные группы пользователей работают только с теми данными, к которым они имеют доступ. При рассмотрении приложений, работающих с одной базой данных, предполагается, что они могут работать параллельно и независимо друг от друга. Именно СУДБ должна обеспечить такую возможность.

**Концептуальный уровень** архитектуры является основным и служит для представления базы данных в общем виде для всех ее приложений и не зависит от них. Концептуальный уровень является промежуточным уровнем в трехуровневой архитектуре и обеспечивает представление всей информации БД в абстрактной форме – это полное представление требований к данным предметной области, независящее от способов их представления и хранения. Описание БД на этом уровне называется концептуальной схемой. Концептуальная схема включает объекты и их атрибуты, связи между объектами, ограничения на данные и т.д.

**Физический уровень** поддерживает представление базы данных в среде хранения. Описание БД на этом уровне называется внутренней схемой или схемой хранения. **Физический уровень** – это собственно данные, расположенные в файлах или страничных структурах, расположенных на внешних носителях. На этом уровне необходимо достичь оптимальной производительности и обеспечить экономное использование дискового пространства. На физическом уровне хранится информация о распределении дискового пространства для хранения данных и индексов, описание подробностей хранения записей, сведения о размещении записей, сведения о сжатии данных и выбранных методах шифрования. Физический уровень контролируется ОСой под руководством СУБД. В соответствии с междууровневыми отображениями принято выделять логическую и физическую независимость данных. Логическая независимость означает защищенность внешних схем от изменений, вносимых в концептуальную схему, кроме того, она предполагает возможность изменения одного приложения без изменения других приложений. Физическая независимость означает защищенность концептуальной схемы от изменений, вносимых в схему хранения. Также она предполагает возможность переноса хранимой информации с одних носителей на другие.

12. Понятие модели данных.

В классической теории БД **модель данных** – это формальная теория представления и обработки данных в СУБД. **Она включает в себя 3 аспекта:**

1. Аспект структуры – методы описания типов и логических структур данных в БД.

2. Аспект манипуляции – методы манипулирования данными.

3. Аспект целостности – методы описания и поддержки целостности БД.

**Модель данных описывает информационные объекты предметной области, взаимосвязи между ними и позволяет:**

1. Определить границу между логистики м и физическим аспектами управления БД.

2. Обеспечить конечным пользователям и программистам возможность и средства общего понимания смысла данных.

3. Определить языковые понятия высокого уровня обеспечивающие возможность выполнения однотипных операций над большими совокупностями записи как единую операцию.

**К числу классических относятся следующие модели данных:**

1. Иерархическая.

2. Сетевая.

3. Реляционная.

В последнее время активно внедряются новые модели данных. Пост-реляционная многомерная и объектно-ориентирована, кроме того, разрабатываются много новых и смешение моделей данных.

13. Теоретико-графовые модели данных.

К этому классу относятся иерархическая и сетевая модель.

Иерархическая модель использует представление предметной области в форме иерархического дерева узоры которого связаны по вертикали отношением предок - потомок. Такая структура предполагает не равноправие между данными - одни данные жёстко подчинены другим. Каждая вершина дерева соответствует сущности предметной области эта сущность характеризуется корректурным количеством атрибутов. Передвижение по дереву всегда начинается с корневой вершины.

В такой структуре между предками и потомками автоматически поддерживается контроль целостности связи основное правило этого контроля: потомок не может существовать без родителя, а у некоторых родителей не может быть потомков. К достоинствам этой модели относятся: эффективное использование памяти ЭВМ и неплохие показатели времянки выполнения.

Эта модель удобна для работы с иерархически упорядоченной информацией. Основным недостатком этой модели являются: дублирование данных её громоздкость для обработки информации со сложными логическими связями, а также сложность понимания для обычного пользователя. Для частичного их преодоления иерархическую модель развили до сетевой в ней любой объект может быть одновременно и главным, и подчинённым и может участвовать в образовании любого количества связей с другими объектами.

**Сетевые модели обладают рядом преимуществ:**

1. Гибкость - можно хранить данные и структура которых намного сложнее простой иерархии.

2. Стандартизация.

3. Быстродействие.

Недостатком сетевой модели является большая сложность, жёсткость схемы базы данных, сложность для понимания, сложность обработки информации и слабые возможности контроля целостности связи.

14. Реляционная модель СУБД.

Относится к теоретико-множественным моделям БД. Является самой популярной на сегодняшний день. Единственная из всех моделей, обеспечивает единообразие представление данных. Реляционная СУБД — это одна из наиболее удачных технологий обработки данных.

В этой модели информационные объекты представляются в виде двумерных таблиц. Каждая строка соответствует экземпляру объекта или конкретному событию или явлению и называется записью. Каждый столбец соответствует конкретному атрибуту объекта, события или явления и называется полем. Все записи одной таблицы имеют одинаковую структуру. Каждое поле имеет конкретный тип данных. Поля должны содержать неделимую (понимаемую только однозначно) информацию. В каждой таблице необходимо наличие хотя бы одного уникального ключа — это поле или набор полей, однозначно идентифицирующий каждую запись. Таких ключей в таблице может быть несколько, тогда один из них определяется как первичный, он должен быть минимально достаточным.

С помощью одной таблицы удобно описывать простейший вид связи между данными – деление одного объекта на множество подобъектов, каждому из которых соответствует запись из таблицы. Поскольку более сложные логические структуры данных в рамках одной таблицы описать невозможно, то применяют связывание таблиц.

Физическое размещение данных в реляционных БД на внешних носителях легко выполняется с помощью обычных файлов. Достоинствами реляционной модели является её простота, понятность и удобство физической и программной реализации на ЭВМ. Основным недостатком является сложность описания иерархический и сетевых связей.

В последних версия реляционных СУБД есть свойство объектно-ориентированных систем. Такие СУБД называют объектно-реляционными.

15. Пост-реляционная модель данных.

Это расширенная реляционная модель которой нет ограничений неделимости данных. То есть она допускает составные многозначные поля. Набор значений многозначного поля считается самостоятельной таблицей, встроенной в основную таблицу. Достоинством этой модели является возможность представления совокупности связанных реляционных таблиц одной пост-реляционной таблицы. Это обеспечивает высокую наглядность представления информации и повышения эффективности её обработки. Недостатком является сложность обеспечения целостности и непротиворечивости хранения данных.

16. Многомерная модель данных.

Развитие информационных систем идёт по двум направлениям: системы оперативной обработки и системы аналитической обработки (они же СППР – системы поддержки принятия решений).

Реляционные системы для аналитической обработки оказались недостаточно гибкими, поэтому были разработаны многомерные системы управления данными. В них данные организованы в виде упорядоченных многомерных массивов. Эти массивы называют гиперкубы. Технология многомерных баз данных реально стала развиваться с середины 1990-х годов. Основные понятия многомерных моделей являются:

1. Измерение – это множество однотипных данных, образующих одну из граней гиперкуба.

2. Мера – это поле, значение которого однозначно определяется фиксированным набором изменений. Однако гиперкуб сам по себе для анализа не пригоден. Для визуализации данных применяются двумерные двоичные табличные изменения. Это называется разрезанием куба.

3. Срез – это подмножество гиперкуба, полученная в результате фиксации одного или нескольких измерений.

Недостатком многомерной системы данных является её громоздкость для простейших задач.

Плюсами можно выделить:

1. Простота системы

2. Относительно низкая стоимость владения, а так бы быстрый возврат инвестиций.

3. В случае использования многомерных СУБД поиск и выборка данных осуществляется значительно быстрее, чем при многомерном взгляде на реляционную базу данных.

4. Многомерные СУБД легко справляются с задачами включения в информационную модель разнообразных встроенных функций.

17. Объектно-ориентированная модель данных.

Объектно-ориентированные СУБД – это результат совмещения возможностей и особенностей баз данных. Большинство объектно-ориентированных представляют собой библиотеку, процедура управления данными которой включаются в прикладную программу. Структура Объектно-ориентированной БД графический представляют в виде дерева, узлами которых являются объекты. Свойства объектов описываются некоторым стандартным типом.

Основным достоинством такой модели является возможность отображения информации о сложных взаимосвязях объектов.

Недостатками являются высокая понятийное сложность, сложности неудобства обработки.

В результате появился гибрид, Объектно- реляционная СУБД. То есть реляционные СУБД, реализующие объекты, классы и наследования.

Основными преимуществами является возможность повторного и совместного использования компания.

Основным недостатком являются сложности и связанные с ней повышенные расходы.

18. продолжение.

Схема отношения – это именованное множество пар (имя атрибута, имя домена). Физический это выглядит как строка заголовков столбцов таблицы. Кортеж – множество пар (имя атрибута, значение). Тогда отношение можно определить как множество картежей, соответствующих одной схеме отношения. Телом отношения называется произвольное множество кортежей.

Первичный ключ – минимальный набор атрибутов, значения которых однозначно определяют кортеж.

19. Свойства отношений.

1. Отсутствие кортежей дубликатов. Для реляционных баз данных это полностью исключает дублирование данных об объектах предметной области.

2. Отсутствие упорядоченности атрибутов, т.е. атрибуты отношений не упорядочены и не пронумерованы. СУБД сама принимает решения о том, в каком физическом порядке хранить значения атрибутов кортежей. Это свойство облегчает выполнение операций модификации схем существующих отношений не только путем добавления новых атрибутов, но и удаления существующих.

3. Атомарность значений атрибутов – атрибут не может быть множеством.

4. Отсутствие упорядоченности кортежей. Физически кортежи хранятся в порядке их ввода. Все сортировки касаются только визуального представления.

20. Реляционная алгебра или алгебра Кодда.

Это замкнутая система операций над отношениями в реляционной модели данных. Реляционные операции, примененные к отношениям таковы, что результат каждой из них также является отношением. Все операции алгебры Кодда можно разделить на теоретико-множественные и специальные реляционные. К первой группе относятся: объединение, пересечение, вычитание, прямое произведение отношений. Ко второй группе относятся: ограничение отношения, проекция отношения, деление отношений, соединение отношений.

Если операция выполняется над одним отношением, то она называется унарной, если над двумя - бинарной.

Бинарные операции можно применять только к совместимым отношениям (совместимость имён атрибутов и типов соответствующих доменов).

21. Индексирование.

Данные сохраняются в таблице в физическом порядке следования записи. При роботе с ним бывает удобно отсортировать их по некоторым полям. Эффективным средством подобной сортировки является использование индексов. Индекс можно представить как специальную структуру в БД, которая позволяет ускорить поиск и сортировку по определённому полю или набору полей таблицы. Кроме того из используют для построения пе первичных или уникальных ключей. Физический индекс - это таблица, которую используют для определения адреса записи. Значения в индексе упорядочены, а сам он относительно мал. Он содержит ключевые значения для каждой записи в таблице данных, а так же уникальные ссылки на соответствующие записи. Кроме того он имеет структуру, оптимизированную под поиск.

Таким образом, первичный ключ, это индекс, позволяющий идентифицировать каждую запись в таблице.

При разработке приложений работающих с БД, чаще всего используются простые индексы, т.е. состоящие из одного поля таблицы. Составной индекс строится на основе значений двух или более полей.

Таблицы в БД могут не иметь индексов, тогда время поиска записи в большой таблице может быть довольно большим. С другой стороны, увеличение количества индексов приводит к замедлению выполнения манипуляций с данными. Для оптимальной производительности БД, индексы создаются на тех полях таблицы, которые часто используются в запросах.

22. Связывание таблиц.

Связи между объектами предметной области в БД отображаются в виде взаимосвязанных таблиц. Организация такой связи называется связыванием. Связи можно установить при создании таблицы и при выполнении приложения. Связи могут быть равноправными и иерархические - (родитель-потомок).

Для связывания используют поля связи - индексированные поля одинакового типа. В подчинённой таблицы для связи с главной задаётся индекс, называемый внешним ключом. Состав полей этого индекса должен совпадать с составом полей индекса главной таблицы.

Выделяют три разновидности связи:

1. Один к одному - используется, если не хотят, что бы таблица разрасталась от второстепенной информации. При этой связи одной записи в радительской таблице соответствует одна запечь в дочерней таблице.

2. Один ко многим - является самой распространённой в реляционных БД, т.к. позволяет моделировать иерархические структуры данных. При этом одной записи родительской таблице может соответствовать несколько записей в дочерней, или наоборот.

3. Многие ко многим - применяется в случаях:

1. Одной записи в родительской таблицы соответствует более одной записи в дочерней.

2. Одной записи в дочерней таблице соответствует много записей в родительской.

Чаще всего, при необходимости подобной связи её разбивают на несколько связей один ко многим.

23. Ссылочная целостность.

Это согласованная связанность между таблицами. Обычно она придерживается комбинированием первичного и внешнего ключа. Для её соблюдения требуется, чтобы любое поле, объявленное внешним ключом могло содержать только значения из поля первичного ключа родительской таблицы.

Ссылочная целостность предотвращает ввод в таблицы несогласованных данных.

Работа со связанными таблицами имеет следующие особенности:

1. При изменении поля связи может нарушится связь между записями в двух таблицах. Поэтому при редактировании поля связи главной таблицы нужно соответственно редактировать и поле связи и его значения всех подчинённых таблиц (каскадное изменение).

2. При удалении записи в главной таблице нужно удалять и соответствующие ей записи в подчинённых таблицах (каскадное удаление).

3. При добавлении записи в подчинённую таблицу, значение её поля связи должно быть установлено равным значению поля связи.

Ограничен6е по этим трём пунктам могут быть наложены на таблицу при их создании или изменении. На ряду с описание полей и индексов они входят в структуру таблицы. Их можно реализовать и программным способом.

24. Принципы поддержки целостности в реляционной базе данных.

Целостность данных подразумевает наличие средств, позволяющих удостоверится, что информация в базе данных всегда остаётся корректной и полной. В реляционной базе целостность можно определить, как соответствие информационной модели предметной области объектом реального мира и их взаимосвязям в каждый момент времени. Ограничение целостности обеспечивают непротиворечивость данных при переводе баз данных из одного состояния в другое и позволяют адекватно отображать предметную область.

Ограничения целостности бывают явные и не явные.

Неявные ограничения определяются самой структурой данных.

Явные ограничения задаются в схеме базы данных с помощью средств описания данных. За выполнение м этих ограничений следит СУБД. Различают так же статические и динамические ограничения целостности.

Статические присущие всем состояниям предметной области, а динамические определяют возможность перехода предметной области из одного состояния в другое.

К средствам обеспечения целостности данных на уровне СУБД относятся:

1. Встроенные средства для назначения первичного ключа, в том числе с автоматическим приращением.

2. Средство поддержания ссылочной целостности, которые обеспечивают запись информации о связях таблиц и автоматический пресекают любую операцию, приводящую к нарушению ссылочной целостности.

Поддержка целостности в реляционной модели данных имеет следующие аспекты:

Во-первых, это поддержка структурной целостности, которая заключается в том, что реляционная СУБД должна допускать работу только с однородными структурами данных типа "реляционное отношение". Оно должно удовлетворять всем ограничениям, накладываемых классической теории реляционной БД.

Реляционная СУБД работает только со структурой данных реляционное отношение по правилам:

1. В таблице не должно быть одинаковых картежей.

2. Столбцы должны соответствовать атрибутам отношения.

3. Должен быть первичный ключ.

4. Каждому атрибуту присваивается уникальное имя.

5. Порядок строк в таблице должен быть произволен.

6. Два отношения, отличающемся только порядком следования столбцов, считаются одинаковыми.

Во-вторых, эта поддержка языковой целостности, которая состоит в том, что реляционная СУБД должна обеспечивать языки описания и манипулирования данными не ниже стандарта SQL. Низкоуровневые средства манипулирования данными, не соответствующие стандарту, не допускаются.

В-третьих, эта поддержка ссылочной целостности, которая означает обеспечение одного из заданных принципов взаимосвязи между экземплярами атрибутов взаимосвязанных таблиц.

Строки подчинённой таблицы уничтожаются при удалении строки основной таблицы.

Строки основной таблицы модифицируются при удалении строки основной таблицы, при этом вместо ключа ставится значение Null.

В процессе модификации данных, ссылочная целостность обеспечивает поддержку непротиворечивого состояния БД. Каждая СУБД должна выполнять структурную, языковую и ссылочную целостность. Они определяют правила работы СУБД с реляционными структурами данных.

25. Достоинства и недостатки реляционной БД.

В настоящее время реляционные СУБД являются наиболее эффективными и распространёнными. Этому способствуют два основных их достоинства:

1. Простое табличное представление, на основе которого можно моделировать наиболее распространённые предметные области. Модель предоставляет средство описания данных на основе только их естественной структуры. Не нужно вводить дополнительные структуры, чтобы получить машинное представление данных. Соответственно она обеспечивает основу языка данных высокого уровня для максимальной независимости программ и машинное представление организации данных.

2. Наличие простого и эффективного математического аппарата, обеспечивающего теоретический базис к организации БД, кроме того он обеспечивает возможность манипулирования данными, не зная их конкретной физической организации во внешней памяти.

Но кроме достоинств реляционная модель имеет недостатки:

1. Ограниченность применения к предметным областям со сложными многоступенчатыми или иерархическими зависимостями.

2. Невозможность адекватного отношения семантики предметной области.

Современные исследования в области пост-реляционных систем просвещены устранению этих недостатков.

26. Задачи проектирования баз данных

Проектирование баз данных состоит в построении комплекса взаимосвязанных моделей данных. Причем процесс построения модели данных неотделим от процессов обработки и манипулирования данными.

Проектирование базы данных состоит из двух основных фаз: логического и физического моделирования.

Хорошо спроектированная база данных:

■ удовлетворяет всем требованиям пользователей к содержимому базы данных.

■ гарантирует непротиворечивость и целостность данных;

■ обеспечивает естественное, легкое для восприятия структурирование информации.

■ удовлетворяет требованиям пользователей к производительности базы данных.

Задача проектировщика состоит в учете всех этих факторов с целью разработки наиболее оптимальной базы данных. Основные задачи проектирования баз данных можно сформулировать следующим образом:

■ обеспечение хранения в БД всей необходимой информации;

■ обеспечение возможности получения данных по всем необходимым запросам;

■ сокращение избыточности и дублирования данных;

■ обеспечение целостности базы данных.

Предметная область — часть реального мира, данные о которой мы хотим отразить в базе данных.

Модель предметной области — это формализованные знания предметной области, выраженные при помощи к Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели.

В общем случае можно выделить следующие этапы проектирования.

1. Системный анализ и словесное описание информационных объектов предметной области.

2. Проектирование концептуальной (инфологической) модели предметной области — частично формализованное описание объектов предметной области в терминах некоторой семантической модели.

3. Даталогическое или логическое проектирование БД, т.е. описание БД в терминах принятой логической модели данных.

4. Физическое проектирование БД, т.е. выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения, каких-либо средств.