Перечень вопросов для подготовки к экзамену:

**. MS SQL Server 2008/2013 и его реализация.**

. OLAP-технология.

. OLTP-технология.

**. Администрирование БД.**

Администрирование базы данных - Управление физической реализацией приложений баз данных: физическое проектирование базы данных и ее реализация, организация поддержки целостности и защиты данных, наблюдение за текущим уровнем производительности системы, а также реорганизация базы данных по мере необходимости. Администратор базы данных (АБД) или Database Administrator (DBA) – это 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. Не менее важной функцией администратора БД является поддержка целостности базы данных.

Основные задачи

Задачи АБД могут незначительно отличаться в зависимости от вида применяемой СУБД, но в основные задачи 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

**. Алгоритм перехода от модели «Сущность-связь» к реляционной модели.**

Шаг 1. Каждая простая сущность превращается в таблицу. Простая сущность - сущность, не являющаяся подтипом и не имеющая подтипов. Имя сущности становится именем таблицы.

Шаг 2. Каждый атрибут становится возможным столбцом с тем же именем; может выбираться более точный формат. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

Шаг 3. Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификатора, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

Шаг 4. Связи многие-к-одному (и один-к-одному) становятся внешними ключами. Т.е. делается копия уникального идентификатора с конца связи "один", и соответствующие столбцы составляют внешний ключ. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

Шаг 5. Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы.

Шаг 6. Если в концептуальной схеме присутствовали подтипы, то возможны два способа:

· все подтипы в одной таблице

· для каждого подтипа - отдельная таблица.

. **Архитектура СУБД Access.**

**. Виды запросов MS SQL Server.**

Описывает типы запросов, которые можно использовать с конструктором запросов и конструктором представлений.

Создание запросов полнотекстового поиска (визуальные инструменты для баз данных)

Описывает, как создать запрос для поиска в полнотекстовом индексированном столбце.

Создание запросов на вставку результатов (визуальные инструменты для баз данных)

Описывает, как создать запрос для вставки результатов инструкции SELECT в таблицу.

Создание запросов на вставку значений (визуальные инструменты для баз данных)

Описывает, как создать запрос для вставки значений в таблицу.

Создание запроса на создание таблицы (визуальные инструменты для баз данных)

Описывает, как создать запрос, создающий новую таблицу.

Создание запросов на удаление (визуальные инструменты для баз данных)

Описывает, как создать запрос, удаляющий данные.

Создание запросов UNION (визуальные инструменты для баз данных)

Описывает, как создать запрос, объединяющий результаты двух или более инструкций SELECT в одном результирующем наборе.

Создание запросов на обновление (визуальные инструменты для баз данных)

Описывает, как создать запрос, обновляющий таблицу.

. **Достоинства и недостатки реляционной модели данных.**

Достоинства

Изложение информации в простой и понятной для пользователя форме (таблица)

Реляционная модель данных основана на строгом математическом аппарате, что позволяет лаконично описывать необходимые операции над данными

Независимость данных от изменения в прикладной программе при изменении

Позволяет создавать языки манипулирования данными не процедурного типа

Для работы с моделью данных нет необходимости полностью знать организацию БД

Недостатки

Для построения запросов и написания прикладных программ нет необходимости знания конкретной организации БД во внешней памяти. Основными недостатками реляционной модели являются следующие: отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей. Относительно низкая скорость доступа и большой объем внешней памяти.

**MS Access. Объекты базы данных и их назначение.**

Объектами в базе данных Access являются таблицы, запросы, формы, отчеты, макросы и модули.

Таблица.Объект, который определяется и используется для хранения и визуализации данных. Каждая таблица обычно содержит информацию об объекте определенного типа. Таблицы связаны между собой, повторяя реальные связи, существующие между объектами. Дополнительно к таблице можно определить несколько индексов для ускорения доступа к данным. В Access объект таблица используется не только для хранения данных, но и для их визуализации, поскольку таблица сама по себе является основным средством представления реляционных данных.

Запрос.Объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Запросы позволяют выбрать только нужные данные, соответствующие определенному критерию. Для создания запроса предоставляется два вида языков: QBE (Query By Example - запрос по образцу) и SQL. Можно создавать запросы на выбор, обновление, удаление или на добавление данных. С помощью запросов существует возможность создавать новые таблицы, используя данные одной или нескольких существующих таблиц и т.д. В Access не делается различия между запросами и таблицами. Соответственно запрос можно основывать и на других запросах, что очень удобно.

Форма.Объект, предназначенный для ввода и отображения данных на экране или управления работой приложения. Таблицы – полезное средство для просмотра и изменения данных, но работать с ними бывает не всегда удобно, а представление данных в них не достаточно наглядно. Для редактирования данных и вывода их на экран в удобном виде предназначены формы. Форма представляет собой бланк, подлежащий заполнению, или маску, накладываемую на набор данных. Формы можно использовать для того, чтобы реализовать требования пользователя к представлению данных из запросов или таблиц. С помощью формы можно красочно оформить данные, представить их в цвете, добавить такие элементы, как поля со списком, комбинированные списки, флажки, кнопки и многое другое. Можно также добавить рисунки и диаграммы и производить вычисления над данными таблиц и запросов.

Формы также предназначены для автоматизации приложения, так как являются средством организации интерфейса в Access. С помощью формы можно в ответ на некоторые события запустить макрос или процедуру, написанную на языке Visual Basic for Application (VBA). С помощью макросов или процедур VBA можно связать несколько форм или отчетов между собой. Например, находясь в одной форме можно открыть другую форму и связать выводимые в них данные таким образом, чтобы при переходе от записи к записи в этой форме в другой форме выводились соответствующие данные.

Отчет.Объект, предназначенный для создания печатного документа. Хотя данные в Access можно вывести на печать в виде таблицы или формы, создание полноценного выходного документа с использованием всех средств форматирования текста и дополнительной обработки данных (например, вычисления промежуточных и окончательных итогов) возможно только с помощью отчетов.

В таблице можно изменять размер шрифта и ширину столбца. В формах есть возможность производить разнообразное форматирование текста, но они плохо приспособлены для проведения сложных вычислений, группировки данных, расчета промежуточных и общих итогов, и на экране показывают чаще всего только одну запись. Отчеты же совмещают достоинства таблицы и формы для печатного представления данных. Кроме вывода на печать в Access допускается экспорт отчета в документ другого приложения, например в Word или Excel.

Макрос.Объект, представляющий собой структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определенное событие.

Модуль.Объект, содержащий программы на VBA, сохраненные под общим именем. Хотя в Access можно без труда создать приложение, состоящее только из таблиц, форм, отчетов и макросов, все же возникает потребность выполнения таких действий, которые нельзя запрограммировать с помощью макросов. Необходимость в модулях появляется также при необходимости обработки ошибок и оформления в виде некоторой процедуры или функции одинаковых вычислений или действий. Модули могут быть независимыми объектами, содержащими функции, которые можно вызывать из любого места приложения, но они могут быть непосредственно «привязаны» к отдельным формам или отчетам для реакции на те или иные происходящие в них события.

**Гипертекстовые и мультимедийные базы данных.**

Суть гипертекстовой технологии заключается в том, что текст представляется как многомерный, т.е. с иерархической структурой типа сети. Материал текста делится на фрагменты. Каждый видимый на экране ЭВМ фрагмент, дополненный многочисленными связями с другими фрагментами, позволяет уточнить информацию об изучаемом объекте и двигаться в одном или нескольких направлениях по выбранной связи. Под гипертекстом понимают систему информационных объектов (статей), объединенных между собой направленными связями, образующими сеть. Каждый объект связывается с информационной панелью экрана, на которой пользователь может выбирать одну из связей. Объекты могут быть текстовыми, графическими, музыкальными, с использованием средств мультипликации, аудио- и видеотехники. Вместо традиционных методов поиска информации по соответствующему поисковому ключу гипертекстовая технология предполагает перемещение от одних объектов информации к другим с учетом их смысловой, семантической связанности.

Базы данных, содержащие мультимедийную информацию, теория относит к базам данных пятого (последнего) поколения. Они получили название мультимедийных баз данных (ММБД) (другое название – мультисредные). Особенности ММБД, вызванные структурной сложностью и неоднородностью хранимой в них информации, показывают, что построение систем баз данных пятого поколения является достаточно сложной задачей, которую преждевременно считать решенной.

Документы 1С: назначение, структура.

Документы 1С: операции над документами.

**Достоинства и недостатки иерархической модели данных.**

К достоинствам иерархической модели данных относятся эффективное использование памяти ЭВМ и неплохие показатели времени выполнения основных операций над данными. Иерархическая модель данных удобна для работы с иерархически упорядоченной информацией. Недостатком иерархической модели является ее громоздкость для обработки информации с достаточно сложными логическими связями, а также сложность понимания для обычного пользователя.

**Достоинства и недостатки сетевой модели данных.**

Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. Недостатком сетевой модели данных являются высокая сложность и жесткость схемы БД, построенной на ее основе. Поскольку логика процедуры выборки данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения.

**Жизненный цикл базы данных.**

Жизненный цикл базы данных (ЖЦБД) – это процесс проектирования, реализации и поддержки базы данных. ЖЦБД состоит из семи этапов:

предварительное планирование

проверка осуществимости

определение требований

концептуальное проектирование

логическое проектирование

физическое проектирование

оценка работы и поддержка базы данных

Журналы документов в 1С: назначение, виды.

**Защита данных в СУБД Access.**

MS Access обеспечивает два традиционных способа защиты базы данных:

установка пароля, требуемого при открытии базы данных;

защита на уровне определения прав пользователей, которая позволяет ограничить возможность получения или изменения той или иной информации в базе данных для конкретного пользователя.

Кроме того, можно удалить изменяемую программу Visual Basic из базы данных, чтобы предотвратить изменения структуры форм, отчетов и модулей, сохранив базу данных как файл MDE.

Гораздо более надежным и гибким способом организации защиты является защита на уровне пользователей. Он подобен способам, используемым в большинстве сетевых систем. Процесс задания защиты на уровне пользователей состоит из двух принципиальных этапов:

создание системы пользователей, объединенных в группы (Сервис > Защита > Пользователи и группы);

задание прав доступа различных пользователей и групп по отношению к объектам базы данных (Сервис > Защита > Разрешения).

**Инструменты управления и утилиты MS SQL Server2008/2013.**

Основное предназначение инструмента, в фигурных скобках указано текущее количество утилит данного типа:

[B]: Работа с бэкапами {17}

[BI]: Business Intelligence Solutions {4}

[D]: Средства по созданию документации {8}

[DA]: Проектирование баз данных {20}

[DC]: Инструменты по сравнению данных {15}

[ETL]: ETL Инструменты {9}

[I]: Встроенные инструменты SQL Server utilities (например, bcp) {15}

[IDX]: Управление индексами {3}

[G]: Генерация тестовых данных {4}

[J]: Управление заданиями (Job) {4}

[M]: Средства мониторинга и оповощений {27}

[MG]: Инструменты для миграции {31}

[MS]: Аналоги SSMS {33}

[REC]: Утилиты восстановления {13}

[S]: Поисковые утилиты {7}

[SC]: Инструменты по сравнению структуры базы данных {21}

[ST]: Утлитиы для работы со статистикой {3}

[ST]: Работа со статистикой {3}

[T]: Инструменты тестирования {7}

[?]: Пока еще не классифицированы{156}

**Инфологическое проектирование базы данных.**

Инфологическое (концептуальное) проектирование – процесс создания внешней (инфологической) модели данных о предметной области, не зависящее от любых физических аспектов ее представления.

На этом этапе используется информация, объединяющая требования пользователей. Инфологическое проектирование базы данных не зависит от таких подробностей ее реализации, как тип выбранной СУБД, набор создаваемых прикладных программ, используемые языки программирования, тип вычислительной системы и т.п. При разработке инфологическая модель постоянно подвергается критической оценке, проверке на соответствие требованиям пользователей, и при необходимости модифицируется. От качества созданной инфологической модели в определяющей степени зависит эффективность конечной базы данных.

Информационные массивы, информационные хранилища: основные понятия, особенности организации.

Кодирование и классификация данных.

Компоненты экономической информационной системы.

Компоненты ЭИС:

Исходная и нормативно-справочная информация

Метод решения задачи, записанной в виде программы, кот. может быть выполнена на ЭВМ

ЭВМ как исполнителя алгоритмов

Пользователей, лиц которые используют результаты решения задачи в своей проф. деятельности

БД

концептуальная схема и инф. процессор, образующие вместе схему хранения и манипулирования данными

**Макросы. Виды макрокоманд. Способы запуска макросов.**

Макрос – программа, состоящая из последовательности макрокоманд. Макрокоманда - это инструкция, ориентированная на выполнение определенного действия над объектами Access и их элементами. Например, макрокоманды можно использовать для выполнения следующих действий:

- для открытия (или закрытия) любой таблицы, запроса, формы или отчета в любом доступном режиме

- для выполнения запроса на выборку или запроса на изменение;

- для выполнения действий в зависимости от значений в базе данных, форме или отчете;

- для запуска других макросов или процедур VBA;

- для применения фильтра и запуска приложений;

Макросы могут быть запущены непосредственно из окна базы данных или окна макроса. Если макрос зависит от события в форме или отчете, его нужно связать с соответствующим событием и запускать при возникновении этого события.

Модели физической организации данных при безфайловой организации.

**Моделирование связей в базах данных.**

Связи создаются с помощью внешних ключей (foreign key).

Внешний ключ — это атрибут или набор атрибутов, которые ссылаются на primary key или unique другой таблицы. Другими словами, это что-то вроде указателя на строку другой таблицы.

Связи делятся на:

Многие ко многим.

Один ко многим.

с обязательной связью;

с необязательной связью;

Один к одному.

с обязательной связью;

с необязательной связью;

**Модель «Сущность-связь». Основные понятия.**

Сущность - это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не некоторого конкретного экземпляра этого типа. Для большей выразительности и лучшего понимания имя сущности может сопровождаться примерами конкретных объектов этого типа.

Связь - это графически изображаемая ассоциация, устанавливаемая между двумя сущностями. Эта ассоциация всегда является бинарной и может существовать между двумя разными сущностями или между сущностью и ей же самой (рекурсивная связь). В любой связи выделяются два конца (в соответствии с существующей парой связываемых сущностей), на каждом из которых указывается имя конца связи, степень конца связи (сколько экземпляров данной сущности связывается), обязательность связи (т.е. любой ли экземпляр данной сущности должен участвовать в данной связи).

**Модель «Сущность-связь». Характеристика связей. Классы принадлежности.**

Связь – некоторая ассоциация между двумя сущностями:

- один-к-одном

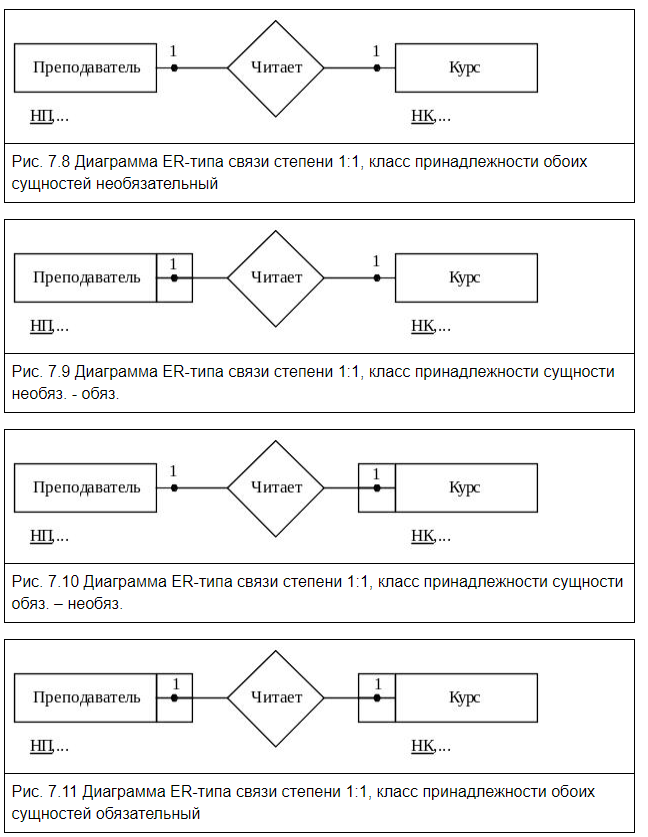
- один-ко-многим

-много-ко-многим

Модальности связи:

- может

- должен



**Назначение запросов. Виды запросов, типы языков запросов СУБД Access.**

Запрос на выборку является наиболее часто используемый. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

Запрос с параметрами — это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные. Можно разработать запрос, выводящий приглашение на ввод нескольких единиц данных, например двух дат. Запросы с параметрами также удобно использовать в качестве основы для форм, отчетов и страниц доступа к данным. Например, на основе запроса с параметрами можно создать месячный отчет о доходах. После ввода месяца Microsoft Access выполняет печать соответствующего отчета.

Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Он подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

Запросы на изменение-запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение.

На удаление записи - удаляет группу записей из одной или нескольких таблиц. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

На обновление записи. вносит общие изменения в группу записей одной или нескольких таблиц, позволяет изменять данные в существующих таблицах.

На добавление записей. добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц.

На создание таблицы. создает новую таблицу на основе всех или части данных из одной или нескольких таблиц, полезен при создании таблицы для экспорта в другие базы данных Microsoft Access или при создания архивной таблицы, содержащей старые записи.

Запросы SQL— это запрос, создаваемый при помощи инструкций SQL. Язык SQL используется при создании запросов, а также для обновления и управления реляционными базами данных, такими как базы данных Microsoft Access.

**Назначение форм базы данных.**

Формы предназначены для ввода и просмотра взаимосвязанных данных БД на экране в удобном виде, который может соответствовать привычному для пользователя документу. Формы можно распечатывать, а также применять для создания панелей управления в приложении.

**Нормализация отношений. Уровни нормализации.**

В первой нормальной форме ER-схемы устраняются повторяющиеся атрибуты или группы атрибутов, т.е. производится выявление неявных сущностей, "замаскированных" под атрибуты.

Во второй нормальной форме устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

**Нормализация. Функциональные зависимости. Аномалии.**

Функциональные зависимости.

Метод нормальных форм является классическим методом проектирования реляционных баз данных. Он основан на зависимости между атрибутами отношений в БД. Дадим определение функциональной зависимости.

Если даны два атрибута X и Y некоторого отношения, то говорят, что Y функционально зависит от X, если в любой момент времени каждому значению X соответствует ровно одно значение Y.

Функциональная зависимость обозначается X -> Y. X и Y могут представлять собой не только единичные атрибуты, но и группы, составленные из нескольких атрибутов одного отношения.

Различают частичную и полную функциональные зависимости. Под частичной функциональной зависимостью понимают зависимость не ключевого параметра от части составного ключа.

Под полной зависимостью понимают зависимость не ключевого параметра от всего составного ключа.

Избыточная функциональная зависимость - зависимость, заключающая в себе такую информацию, которая может быть получена на основе других зависимостей, имеющихся в базе данных.

Корректной считается такая схема базы данных, в которой отсутствуют избыточные функциональные зависимости. В противном случае приходится прибегать к процедуре декомпозиции (разложения) имеющегося множества отношений. При этом порождаемое множество содержит большее число отношений, которые являются проекциями отношений исходного множества. Обратимый пошаговый процесс замены данной совокупности отношений другой схемой с устранением избыточных функциональных зависимостей называется нормализацией.

**Общая характеристика современных средств разработки файл – серверных приложений.**

В индустрии СУБД для персональных компьютеров отразились тенденции нормализации систем (Rightsizing). В последнее время в этой области происходили два встречных процесса: (1) разукрупнение серверов БД - появление новых версий серверов БД Informix, Oracle и т.д. сначала в варианте для рабочих групп, а потом облегченные версии для одиночных персональных компьютеров; (2) укрупнение СУБД для персональных компьютеров - новые "персональные" СУБД и связанные с ними инструментальные средства развивались в сторону "истинно реляционных" СУБД, т.е. серверов БД, приложений клиент-сервер и инструментальных средств программирования 4GL и быстрой разработки RAD.

Новые СУБД для персональных компьютеров и соответствующие инструментальные средства разработки файл-серверных приложений обладают перечисленными ниже общими чертами.

Визуальный характер программирования приложений особенно в части создания диалогового графического интерфейса пользователя. Это множество поддерживаемых диалоговых объектов, поддержка механизма drag-and-drop и наличие мастеров, помогающих реализовать сложные процедуры.

Управляемость приложений в соответствии с событиями диалога и обеспечение доступа к БД позволяет строить гибкий интерфейс пользователя и поддерживать ссылочную целостность БД.

Встроенная поддержка языка структурированных запросов SQL (Standard Query Language) закладывает возможность масштабирования создаваемых файл-серверных приложений до уровня приложений клиент-сервер.

Имеется возможность построения приложений клиент-сервер за счет реализации доступа к серверам БД напрямую или через интерфейс ODBC для открытого взаимодействия с базами данных.

Использование объектно-ориентированного языка разработки приложений (по крайней мере в части диалога) позволяет широко использовать механизм наследования и тем самым использовать ранее произведенные программные компоненты.

Поддержка компонентно-ориентированного программирования дает возможность расширения приложений за счет использования готовых внешних визуальных объектов типа VBX и OCX (ActiveX).

"Истинно реляционная" база данных представляет собой объединенный набор файлов, содержащий таблицы, индексы и т.п., что облегчает сопровождение БД и приложений и является основой для поддержки целостности данных.

Поддерживается общий для информационной системы словарь данных (data dictionary), который содержит описание структуры БД, типы полей, правила поддержки ограничений целостности и т.п.

Поддержка целостности БД (данных, ссылок и транзакций) позволяет создавать приложения с необходимым уровнем надежности и сохранности данных.

Возможности серверных процедур обработки (триггеров и хранимых процедур) закладывают основу для масштабирования приложений, позволяют гибко распределять прикладную логику между клиентом и сервером при переходе к архитектуре клиент-сервер.

Хранение в БД описания проекта создаваемого приложения является прообразом репозитория инструментальных средств быстрой разработки RAD и CASE-систем.

**Ограничения базовых моделей данных.**

К внутренним ограничениям целостности в нотациях IDEF1X можно отнести:

1) ограничения на значения ключевых атрибутов: уникальность значений атрибутов первичных и альтернативных ключей определяется понятием ключа. Эти ограничения задаются соответствующим выделением ключей в концептуальной схеме (PRIMARY KEY, UNIQUE);

2) ограничения на допустимые значения атрибутов: все допустимые значения атрибутов должны удовлетворять условию принадлежности соответствующему домену. Эти ограничения задаются соответствующим описанием доменов и атрибутов. Кроме того, в описании атрибутов указывается обязательность значений определенных атрибутов (NULL, NOT NULL);

3) ограничения на существующие значения (ссылочные ограничения): существование одних сущностей (дочерних, типа категория) ставится в зависимость от существования других (родительских, родовых)

**Основные понятия баз данных.**

**Информационная система** – это совокупность программно-аппаратных средств, способов и людей, которые обеспечивают сбор, хранение, обработку и выдачу информации для решения поставленных задач. На ранних стадиях использования информационных систем применялась файловая модель обработки. В дальнейшем в информационных системах стали применяться базы данных. Базы данных являются современной формой организации, хранения и доступа к информации. Примерами крупных информационных систем являются банковские системы, системы заказов железнодорожных билетов и т.д.

**База данных** – это интегрированная совокупность структурированных и взаимосвязанных данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных. Обычно база данных создается для предметной области.

**Предметная область** – это часть реального мира, подлежащая изучению с целью создания базы данных для автоматизации процесса управления.

Наборы принципов, которые определяют организацию логической структуры хранения данных в базе, называются **моделями данных**.

Существуют **4 основные модели данных** – списки (плоские таблицы), реляционные базы данных, иерархические и сетевые структуры.

В течение многих лет преимущественно использовались плоские таблицы (плоские БД) типа списков в Excel. В настоящее время наибольшее распространение при разработке БД получили реляционные модели данных. Реляционная модель данных является совокупностью простейших двумерных таблиц – **отношений**(англ. relation),т.е. простейшая двумерная таблица определяется как **отношение**(множество однотипных записей объединенных одной темой**)**.

**Основные требования к структурам реляционной базы данных.**

1) каждая таблица должна иметь уникальное имя;

2) столбцы одной таблицы должны иметь уникальные имена, поэтому порядок следования столбцов в таблице не имеет значения;

3) каждая строка таблицы должна быть уникальной, т.е. в одной таблице не может быть двух одинаковых строк;

4) в каждой ячейке таблицы может быть только одно значение;

5) в идеале каждое данное должно храниться в базе в единственном экземпляре, т.е. не должно быть избыточности и дублирования данных. На практике избыточность данных должна быть сведена к разумному минимуму;

6) в базе не должны содержаться противоречивые данные, что достигается на практике обеспечением целостности данных.

Особенности работы с типами данных xml.

**Первичные и внешние ключи.**

Первичный ключ - это ключ, который помогает однозначно идентифицировать кортеж базы данных, тогда как Внешний ключ - это ключ, который используется для идентификации связи между таблицами с помощью первичного ключа одной таблицы, который является первичным ключом, одна таблица действует как внешний ключ к другой таблице.

Планирование конфигурации дисков. RAID контроллеры.

**Пользователи баз данных.**

Пользователи баз данных -- это специальные объекты, которые создаются на уровне базы данных и используются для предоставления разрешений в базе данных (на таблицы, представления, хранимые процедуры). Для пользователей используется термин database users(или просто users), в отличие от логинов (logins) -- учетных записей для подключения к SQL Server.

**Понятие и основные свойства транзакций.**

Транзакция- это последовательность операторов манипулирования данными, выполняющаясякак единое целое(все или ничего) и переводящая базу данныхиз одного целостного состояния в другое целостное состояние.

Транзакция обладает четырьмя важными свойствами, известными как свойства АСИД:

(А) Атомарность. Транзакция выполняется как атомарная операция - либо выполняется вся транзакция целиком, либо она целиком не выполняется.

(С) Согласованность. Транзакция переводит базу данных из одного согласованного (целостного) состояния в другое согласованное (целостное) состояние. Внутри транзакции согласованность базы данных может нарушаться.

(И) Изоляция. Транзакции разных пользователей не должны мешать друг другу (например, как если бы они выполнялись строго по очереди).

(Д) Долговечность. Если транзакция выполнена, то результаты ее работы должны сохраниться в базе данных, даже если в следующий момент произойдет сбой системы.

**Понятие ключей и индексов.**

Ключи - это идентификатор конкретных столбцов, определяющий их специфические возможности. Первичный ключ и прочие. Индекс же - разновидность таблицы, служащая для упрощения поиска по определённой выдаче.

**Понятие предметной области.**

Предметная область - это совокупность реальных объектов (сущностей), которые представляют интерес для пользователей. Объект (сущность) - предмет, процесс или явление, о котором собирается информация, необходимая для решения задачи. Объектом может быть человек, предмет, событие. Каждый объект характеризуется рядом основных свойств - атрибутов. Атрибутом называется поименованная характеристика объекта.

Постреляционные системы.

Программные модули 1С. Виды программных модулей.

Протокол ODBC и его реализация.

**Процесс прохождения запроса пользователя к базе данных.**

1. Пользователь посылает СУБД запрос на получение данных из БД.
2. Анализ прав пользователя и внешней модели данных, соответствующей данному пользователю, подтверждает или запрещает доступ данного пользователя к запрошенным данным.
3. В случае запрета на доступ к данным СУБД сообщает пользователю об этом (стрелка 12) и прекращает дальнейший процесс обработки данных, в противном случае СУБД определяет часть концептуальной модели, которая затрагивается запросом пользователя.
4. СУБД запрашивают информацию о части концептуальной модели.
5. СУБД получает информацию о запрошенной части концептуальной модели.
6. СУБД запрашивает информацию о местоположении данных на физическом уровне (файлы или физические адреса).
7. В СУБД возвращается информация о местоположении данных в терминах операционной системы.
8. СУБД вежливо просит операционную систему предоставить необходимые данные, используя средства операционной системы.
9. Операционная система осуществляет перекачку информации из устройств хранения и пересылает ее в системный буфер.
10. Операционная система оповещает СУБД об окончании пересылки.
11. СУБД выбирает из доставленной информации, находящейся в системном буфере, только то, что нужно пользователю, и пересылает эти данные в рабочую область пользователя.

**Реализация файл – серверных приложений.**

В отличии от централизованной системы архитектура "файл-сервер" (таблица 3.1 и рисунок 3.1) не имеет сетевого разделения компонентов диалога PS и PL, использует ПК для функций отображения, что облегчает построение графического интерфейса. Файл-сервер только извлекает данные из файлов, так что дополнительные пользователи и приложения добавляют лишь незначительную нагрузку на ЦП. Каждый новый клиент добавляет вычислительную мощность к сети.

Режимы работы с БД: модели серверов БД.

Режимы работы с БД: модель сервера БД.

Режимы работы с БД: модель удаленного доступа к данным.

Режимы работы с БД: модель удаленного управления данными.

Режимы работы с БД: модель файлового сервера.

Режимы работы с БД: трехуровневые модели.

**Реляционная алгебра. Операции: выборка, проекция**

Проекция является операцией, при которой из отношения выделяются атрибуты только из указанных доменов, то есть из таблицы выбираются только нужные столбцы, при этом, если получится несколько одинаковых кортежей, то в результирующем отношении остается только по одному экземпляру подобного кортежа.

Для примера сделаем проекцию на таблице PRODUCTS выбрав из нее ID и PRICE.

Выборка — это операция, которая выделяет множество строк в таблице, удовлетворяющих заданным условиям. Условием может быть любое логическое выражение.

Для примера сделаем выборку из таблицы с ценой больше 90.

**Реляционная алгебра. Операции: естественное соединение, условное соединение, деление**

Естественным соединением (англ. Natural join) двух отношений R1

и R2 называется отношение, в котором заголовок является объединением заголовков R1и R2, а тело состоит из кортежей, полученных всевозможными соединениями кортежей R1и R2, имеющих равные значения одноимённых атрибутов. Обозначение: R1⋈R2

Условным соединением (англ. Conditional join) двух отношений R1 и R2, у которых нет общих атрибутов, по условию θ называется отношение, в котором заголовок является объединением заголовков R1 и R2, а кортежами тела являются всевозможные конкатенации кортежей тел R1 и R2, удовлетворяющих условию θ. Обозначение: R1×θR2

**Реляционная алгебра. Операции: объединение, пересечение, разность, декартово произведение**

Умножение или декартово произведение является операцией, производимой над двумя отношениями, в результате которой мы получаем отношение со всеми доменами из двух начальных отношений. Кортежи в этих доменах будут представлять из себя все возможные сочетания кортежей из начальных отношений. На примере будет понятнее.

Операция соединения обратна операции проекции и создает новое отношение из двух уже существующих. Новое отношение получается конкатенацией кортежей первого и второго отношений, при этом конкатенации подвергаются отношения, в которых совпадают значения заданных атрибутов. В частности, если соединить отношения PRODUCTS и SELLERS, этими атрибутами будут атрибуты доменов ID.

Также для понятности можно представить соеднинение как результат двух операций. Сначала берется произведение исходных таблиц, а потом из полученного отношения мы делаем выборку с условием равенства атрибутов из одинаковых доменов. В данном случае условием явлется равенство PRODUCTS.ID и SELLERS.ID.

Результатом операции пересечения будет отношение, состоящее из кортежей, полностью входящих в состав обоих отношений.

Результатом вычитания будет отношение, состоящее из кортежей, которые являются кортежами первого отношения и не являются кортежами второго отношения.

**Реляционная модель данных.**

Реляционная модель представляет собой совокупность данных, состоящую из набора двумерных таблиц. В теории множеств таблице соответствует термин отношение (relation), физическим представлением которого является таблица, отсюда и название модели– реляционная. Реляционная модель является удобной и наиболее привычной формой представления данных. При табличной организации данных отсутствует иерархия элементов.

**Свойства отношений.**

Рассмотрим теперь свойства отношений, которые следуют из приведенного выше определения отношения. В любом отношении

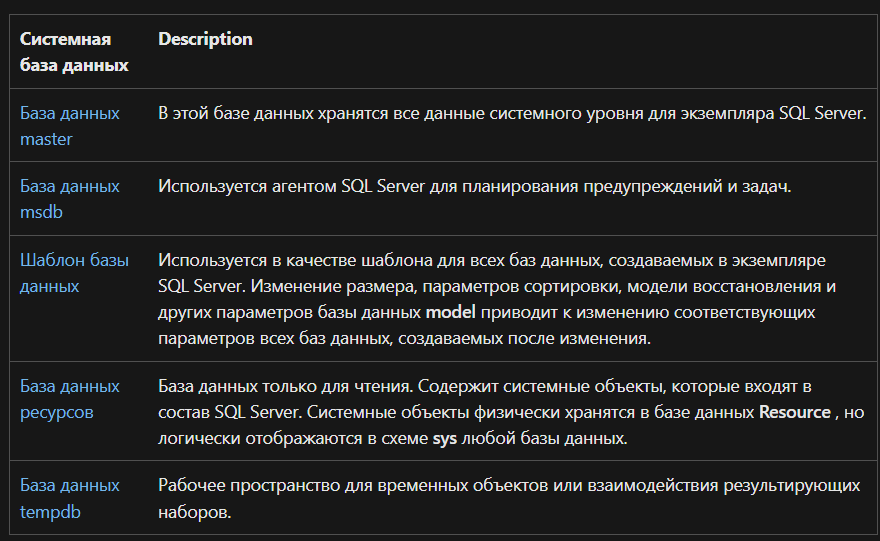
Отсутствуют одинаковые кортежи

Отсутствует упорядоченность кортежей

Отсутствует упорядоченность атрибутов

Все значения атрибутов атомарные

**Системные базы данных MS SQL Server 2008/2013.**



Системный анализ предметной области.

С точки зрения проектирования БД в рамках системного анализа, необходимо осуществить первый этап, то есть провести подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами.

В общем случае существуют два подхода к выбору состава и структуры предметной области:

Функциональный подход – реализует принцип движения «от задач» и применяется тогда, когда заранее известны функции некоторой группы лиц и комплексов задач, для обслуживания информационных потребностей которых создается рассматриваемая БД. В этом случае мы можем четко выделить минимальный необходимый набор объектов предметной области, которые должны быть описаны.

Предметный подход – когда информационные потребности будущих пользователей БД жестко не фиксируются. Они могут быть многоаспектными и весьма динамичными. Мы не можем точно выделить минимальный набор объектов предметной области, которые необходимо описывать. В описание предметной области в этом случае включаются такие объекты и взаимосвязи, которые наиболее характерны и наиболее существенны для нее. БД, конструируемая при этом, называется предметной, то есть она может быть использована при решении множества разнообразных, заранее не определенных задач.

Чаще всего на практике рекомендуется использовать некоторый компромиссный вариант, который, с одной стороны, ориентирован на конкретные задачи или функциональные потребности пользователей, а с другой стороны, учитывает возможность наращивания новых приложений.

Системный анализ должен заканчиваться подробным описанием информации об объектах предметной области, которая требуется для решения конкретных задач и которая должна храниться в БД.

Современные технологии баз данных и СУБД.

**Способы создания баз данных в MS SQL Server 2008/2013.**

В обозревателе объектов подключитесь к экземпляру компонента Компонент SQL Server Database Engine и разверните его.

Щелкните правой кнопкой мыши узел Базы данных и выберите команду Создать базу данных.

В поле Новая база данных введите имя базы данных.

Чтобы создать базу данных, приняв все значения по умолчанию, нажмите кнопку ОК; в противном случае продолжайте выполнять указанные ниже дополнительные действия.

Чтобы изменить имя владельца, нажмите ( … ) и выберите другого владельца.

Чтобы создать базу данных, нажмите кнопку ОК.

Использование Transact-SQL

Создание базы данных

Установите соединение с компонентом Компонент Database Engine.

На стандартной панели выберите пункт Создать запрос.

Скопируйте приведенный ниже пример в окно запроса и нажмите кнопку Выполнить. В этом примере создается база данных Sales. Так как ключевое слово PRIMARY не используется, первый файл (Sales\_dat) становится основным файлом. Так как MB или KB не указаны в параметре SIZE для Sales\_dat файла, он использует MB и выделяется в мегабайтах. Резервную копию базы данных Sales\_log выделено в мегабайтах, потому что суффикс MB явно указан в параметре SIZE .

SQL

Копировать

USE master;

GO

CREATE DATABASE Sales ON

(NAME = Sales\_dat,

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\saledat.mdf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 5)

LOG ON

(NAME = Sales\_log,

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\salelog.ldf',

SIZE = 5 MB,

MAXSIZE = 25 MB,

FILEGROWTH = 5 MB);

GO

Сравнительный анализ реляционных СУБД. https://drach.pro/blog/hi-tech/item/196-popular-relational-dbms-2022

Структура базы данных 1С.

Структура оператора Select.

СУБД Access. Запуск запросов из формы. Задание условий отбора через поле со списком в форме.

СУБД Access. Построение схемы данных. Ссылочная целостность.

СУБД Access. Создание пользовательского интерфейса с базой данных.

СУБД Access. Создание форм. Главная и подчиненная формы.

СУБД Access. Таблицы. Задание ограничений целостности, определяемых пользователем.

Типовые элементы управления для построения и редактирования объектов.

Типы данных 1С.

Типы данных MS Access.

Трехуровневая архитектура баз данных.

Управление санкционированным доступом к данным в БД.

Физические модели данных: индексные файлы.

Физические модели данных: классификация файловых структур.

Формы в 1С: назначение, виды, структура.

Функции администраторов баз данных.

Функции СУБД.

Функции СУБД – это: описание данных, манипулирование данными, использование данных. Данные функции реализуются благодаря наличию у СУБД языка описания данных (ЯОД), языка манипулирования данными (ЯМД) и языка запросов.

Целостность данных, понятие и способы обеспечения.

Целостность данных — это поддержание и обеспечение точности и согласованности данных на протяжении всего их жизненного цикла и является критическим аспектом для проектирования, реализации и использования любой системы, которая хранит, обрабатывает или извлекает данные

Элементы макета формы.

Элементы пользовательского интерфейса MS Access.

Этапы проектирования баз данных.

Этапы проектирования и создания базы данных определяются следующей последовательностью:

построение информационно-логической модели данных предметной области

определение логической структуры реляционной базы данных

конструирование таблиц базы данных

создание схемы данных

ввод данных в таблицы (создание записей)

разработка необходимых форм, запросов, макросов, модулей, отчетов

разработка пользовательского интерфейса

Этапы развития баз данных.

Язык QBE.

Язык SQL.