**ВОПРОСЫ К ЭКЗАМЕНУ ПО ПРЕДМЕТУ «БАЗЫ И БАНКИ ДАННЫХ»**

[**Вопрос 4. Обследование объекта автоматизации: методологии моделирования** 2](#_Toc71021270)

[**Вопрос 5. Структурно-функциональный подход** 2](#_Toc71021271)

[**Вопрос 6. Объектно-ориентированный подход** 3](#_Toc71021272)

[**Вопрос 7. Интегрированный подход** 4](#_Toc71021273)

[**Вопрос 8. Организационная структура** 5](#_Toc71021274)

[**Вопрос 9. Виды организационных структур** 6](#_Toc71021275)

[**Вопрос 10. Дивизиональная организационная структура** 7](#_Toc71021276)

[**Вопрос 11. Матричная организационная структура** 8](#_Toc71021277)

[**Вопрос 12. Документы, определяющие организационную структуру предприятия** 10](#_Toc71021278)

[**Вопрос 13. Функциональная модель объекта автоматизации** 12](#_Toc71021279)

[**Вопрос 14. Влияние функциональной структуры объекта автоматизации на структуру автоматизированной системы** 15](#_Toc71021280)

[**Вопрос 15. Информационная структура и информационная модель объекта автоматизации** 16](#_Toc71021281)

[**Вопрос 16. Документ и его структура** 19](#_Toc71021282)

[**Вопрос 17. Учет документооборота в организации. Жизненный цикл документа.** 22](#_Toc71021283)

[**Вопрос 19. Бизнес-процесс и его элементы** 25](#_Toc71021284)

[**Вопрос 20. Проектирование процессной модели ОА**. **Нотация eEPC (Extended Events Process Chain) для проектирования процессной модели** 26](#_Toc71021285)

[**Вопрос 21. Технология проектирования баз данных** 32](#_Toc71021286)

[**Вопрос 22. Концептуальное проектирование базы данных. Модель «Сущность - Связь»** 34](#_Toc71021287)

[**Вопрос 22. Основные понятия методологии «Сущность - Связь»** 35](#_Toc71021288)

[**Вопрос 23. Систематизация предметной области** 37](#_Toc71021289)

[**Вопрос 24. Абстракция обобщения. Абстракция агрегации.** 38](#_Toc71021290)

[**Вопрос 25. Проектирование концептуальной модели** 39](#_Toc71021291)

[**Вопрос 26. Логическое моделирование базы данных.** 43](#_Toc71021292)

[**Вопрос 27. Основные понятия логической модели** 44](#_Toc71021293)

[**Вопрос 28. Построение отношений на основе сущностей КМ.** 49](#_Toc71021294)

[**Вопрос 29. Нормализация логической модели БД** 51](#_Toc71021295)

[**Вопрос 30. Определение механизмов целостности данных** 54](#_Toc71021296)

[**Вопрос 31. Физическая модель базы данных** 56](#_Toc71021297)

# **Вопрос 4. Обследование объекта автоматизации: методологии моделирования**

Понимание того, как работает та или иная система, часто сводится к созданию ее модели. Таким образом, чтобы создать автоматизированную систему или базу данных, как нижний уровень автоматизированной системы, необходимо разработать модель, адекватно описывающую данный объект автоматизации.   
**Модель** – это абстрактное представление реальности в какой-либо форме (математической, физической, символической, графической или дескриптивной), предназначенное для представления определённых аспектов этой реальности и позволяющее получить ответы на изучаемые вопросы.   
**Объектные методики** рассматривают моделируемую область как набор взаимодействующих объектов. Объект определяется как осязаемая реальность – предмет или явление, имеющие четко определяемое поведение. Целью применения данной методики является выделение объектов, составляющих организацию, и распределение между ними ответственностей за выполняемые действия.

**Функциональные методики** рассматривают предметную область как набор функций, преобразующий поступающий поток информации в выходной поток. Процесс преобразования информации потребляет определенные ресурсы. Основное отличие от объектной методики заключается в четком отделении функций (методов обработки данных) от самих данных.

Каждый из представленных подходов обладает своими преимуществами. Объектный подход лучше соответствует существующим структурам организации. Функциональное моделирование хорошо показывает себя в тех случаях, когда организационная структура находится в процессе изменения или вообще слабо оформлена. Подход от выполняемых функций интуитивно лучше понимается исполнителями при получении от них информации об их текущей работе.

# 

# **Вопрос 5. Структурно-функциональный подход**

Согласно данному подходу сама объект автоматизации рассматривается как функция (или функциональный блок), размещаемый в центре схемы, в который входят и выходят 4 вида потоков, представляющие четыре вида обмена объекта автоматизации, как системы или функции, с внешней средой:

**Вход** – состоит из ресурсов, информации, задач, поступающих в организацию из внешней среды

**Выход** - состоит из ресурсов, информации, задач, поступающих во внешнюю среду из организации

**Управление** (Контроль и ограничения) – информационные процессы, ресурсы и пр., регулирующие деятельность организации

**Механизмы** (Используемые ресурсы) – ресурсы, за счет которых предприятие осуществляет свою деятельность (функцию)

# **Вопрос 6. Объектно-ориентированный подход**

Объектно-ориентированная модель предназначена для создания моделей систем с целью их последующей реализации в виде объектно-ориентированных программ. Главным структурообразующим элементом данных методологий является объект, который объединяет данные и процедуры.   
В модели объекта автоматизации – это участники бизнес-процесса (активные объекты) и пассивные объекты (материалы, документы), над которыми выполняют действия активные объекты. Наиболее распространенной объектно-ориентированной методологией является методология UML.

# 

# **Вопрос 7. Интегрированный подход**

Интегрированная модель — это взаимоувязанная совокупность нескольких частных моделей (организационной, функциональной, информационной, ресурсов, бизнес-процессов), каждая из которых описывает отдельные перспективы его структуры, а все вместе они образуют полное и комплексное представление о динамике его исполнения.

ARIS позволяет отражать в единой интегрированной модели «Дом ARIS»: оргструктуры, функции, данные, процессы.

# 

# **Вопрос 8. Организационная структура**

Моделирование организационной структуры компании – стартовая точка в создании топологии компьютерной сети и подсистем АСУ, т.к. определяет участников бизнес-процесса, а также пользователей проектируемой информационной системы

**Организационная структура** (оргструктура) – совокупность подразделений организации и их взаимосвязей, в рамках которой между подразделениями распределяются функциональные задачи, определяются полномочия и ответственность руководителей и должностных лиц.

Организационная структура, как правило, устанавливается исходя из объема и содержания задач, решаемых в организации, направленности и интенсивности сложившихся на предприятии информационных и документационных потоков и с учетом его организационных и материальных возможностей.

Организационная структура используется для определения иерархии внутри организации. Эта структура разрабатывается для определения того, как работает организация. Организационная структура иллюстрируется, как правило, с использованием организационной схемы, которая также называется органограммой или органиграммой.

Организационная структура объекта автоматизации напрямую влияет на состав пользовательских ролей, но и на состав подсистем, входящих автоматизированную систему обработки информации для объекта автоматизации.

# 

# **Вопрос 9. Виды организационных структур**

Оргструктура формируется в зависимости от целей и задач организации.

**Линейно-функциональная оргструктура** известна как традиционная или классическая, поскольку она была первой структурой, подвергшейся изучению и разработке. Построена на основе иерархии подразделений, каждое из которых осуществляет свои функции. Линейно-функциональную структуру целесообразно использовать в тех организациях, которые выпускают относительно ограниченную номенклатуру продукции, действуют в стабильных внешних условиях и для обеспечения своего функционирования требуют решения стандартных управленческих задач.

**Преимущества**

Стимулирует профессиональную специализацию

Уменьшает дублирование усилий и потребление материальных ресурсов

Улучшает координацию внутри функциональной области

**Недостатки**

Элементы такой структуры не заинтересованы в достижении целей и решении задач организации и стремятся лишь выполнять свою часть работы.

# 

# **Вопрос 10. Дивизиональная организационная структура**

Дивизионная структура возникает тогда, когда в качестве основного критерия объединения сотрудников по отделам выступает продукция, выпускаемая организацией. Строится по одному из признаков: по продукту, по группам пользователей, по географическому расположению.

В дивизиональной структуре различия мнений между отделами будут урегулированы на уровне подразделения-дивизиона, а не руководителя компании.

**Плюсы:**

высокая степень самостоятельности дивизионов;

высокая степень выживаемости в условиях современного рынка;

развитие у управляющих дивизионов предпринимательских навыков.

**Минусы:**

появление дублирующих функций в дивизионах:

ослабление связей между сотрудниками различных дивизионов;

частичная потеря контроля над деятельностью дивизионов;

отсутствие одинакового подхода к управлению различными дивизионами

# 

# **Вопрос 11. Матричная организационная структура**

Матричная оргструктура формируется по паре функция+продукт. Примером матричной организационной структуры является проектная организация, функционирующая следующим образом: при запуске новой программы назначается Ответственный руководитель, который ведет ее от начала и до конца. Из специализированных подразделений ему для работы выделяют необходимых сотрудников, которые по завершении реализации возложенных на них задач возвращаются обратно в свои структурные подразделения.

**Плюсы:**

возможность оперативной ориентации на потребности своих клиентов;

значительное сокращение времени на внедрение различных новшеств;

своеобразная кузница руководящих кадров, так как руководителем проекта может быть назначен практически любой сотрудник предприятия.

**Минусы:**

опасность возникновения конфликтов между руководителями проектов и начальниками подразделений, из которых они получают специалистов для реализации своих проектов;

большая сложность в управлении и координации деятельности организации в целом.

# 

# **Вопрос 12. Документы, определяющие организационную структуру предприятия**

Организационная структура предприятия определяется следующими документами:

**Устав организации**

Устав — это учредительный документ, являющийся обязательным для юридических лиц. В документе указывается информация о различных юридических фактах, связанных с компанией: состав учредителей, местоположение фирмы, размер уставного капитала, порядок распределения прибыли и т.д.

**Органиграмма**

Организационная структура - документ, устанавливающий количественный и качественный состав подразделений предприятия и схематически отражающий порядок их взаимодействия между собой. Структура предприятия устанавливается исходя из объема и содержания задач, решаемых предприятием, направленности и интенсивности, сложившихся на предприятии информационных и документационных потоков и с учетом его организационных и материальных возможностей. В организационной структуре отражаются все подразделения предприятия, порядок их подчиненности. Можно также отразить как вертикальные, так и горизонтальные связи между подразделениями.

**Штатное расписание** – организационно-правовой документ, устанавливающий количественный и качественный состав работников предприятия в целом и по каждому из его структурных подразделений. Штатное расписание определяет структуру, численность и должностной состав работников предприятия с указанием должностных окладов.

**Положение о структурном подразделении -**

это правовой акт, устанавливающий статус, функции, права, обязанности и ответственность структурных подразделений.

При формировании структуры предприятия необходимо не только определить функции каждой структурной единицы, но и закрепить их в соответствующем документе.

**Должностная инструкция** - правовой акт, издаваемый предприятием в целях регламентирования организационно-правового положения работника, его обязанностей, прав, ответственности и обеспечивающий условия для его эффективной работы. Должностная инструкция, являясь важным организационно-правовым документом, должна включать перечень функциональных обязанностей работника, квалификационные требования для замещения данной должности, определять степень его ответственности и компетентности.

# 

# **Вопрос 13. Функциональная модель объекта автоматизации**

Функция – это предметно-ориентированное задание или действие, выполняемое над объектом, в результате которых достигается одна или несколько целей, стоящих перед компанией.

Функции предприятия определены в *Уставе предприятия*. Функции предприятия распределяются по компонентам организационной структуры предприятия и определяются в документах *Положение о подразделении, Должностная инструкция, Должностные обязанности* и т.д*.*

При обследовании объекта автоматизации исследуется и его функциональная структура (или функциональная модель).

Функциональная модель представляется через "дерево" основных функций, реализуемых на предприятии. Модель строится иерархически — от верхнего уровня функций к нижнему (через декомпозицию).

Функции могут быть описаны с различными уровнями детализации. На самом верхнем уровне описываются наиболее сложные функции, представляющие собой отдельные процессы либо их последовательности.

**Основные процессы –** все, что связано с непосредственной деятельностью организации, предприятия, компании, что создает прибавочную стоимость: производство, закупки, хранение, реализация и т.д.

**Вспомогательные процессы -** все, что связано с обслуживанием основных процессов: административно-хозяйственное обеспечение, управление персоналом, ИТ-инфраструктура и т.д.

**Управление компанией -** Финансово-аналитические структуры, высшие управленческие инфраструктуры компании, занимающиеся долгосрочным планированием, прогнозированием и выбором стратегий.

ARIS — методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций.

Любая организация в методологии ARIS рассматривается с пяти точек зрения: организационной, функциональной, обрабатываемых данных, структуры бизнес-процессов, продуктов и услуг.

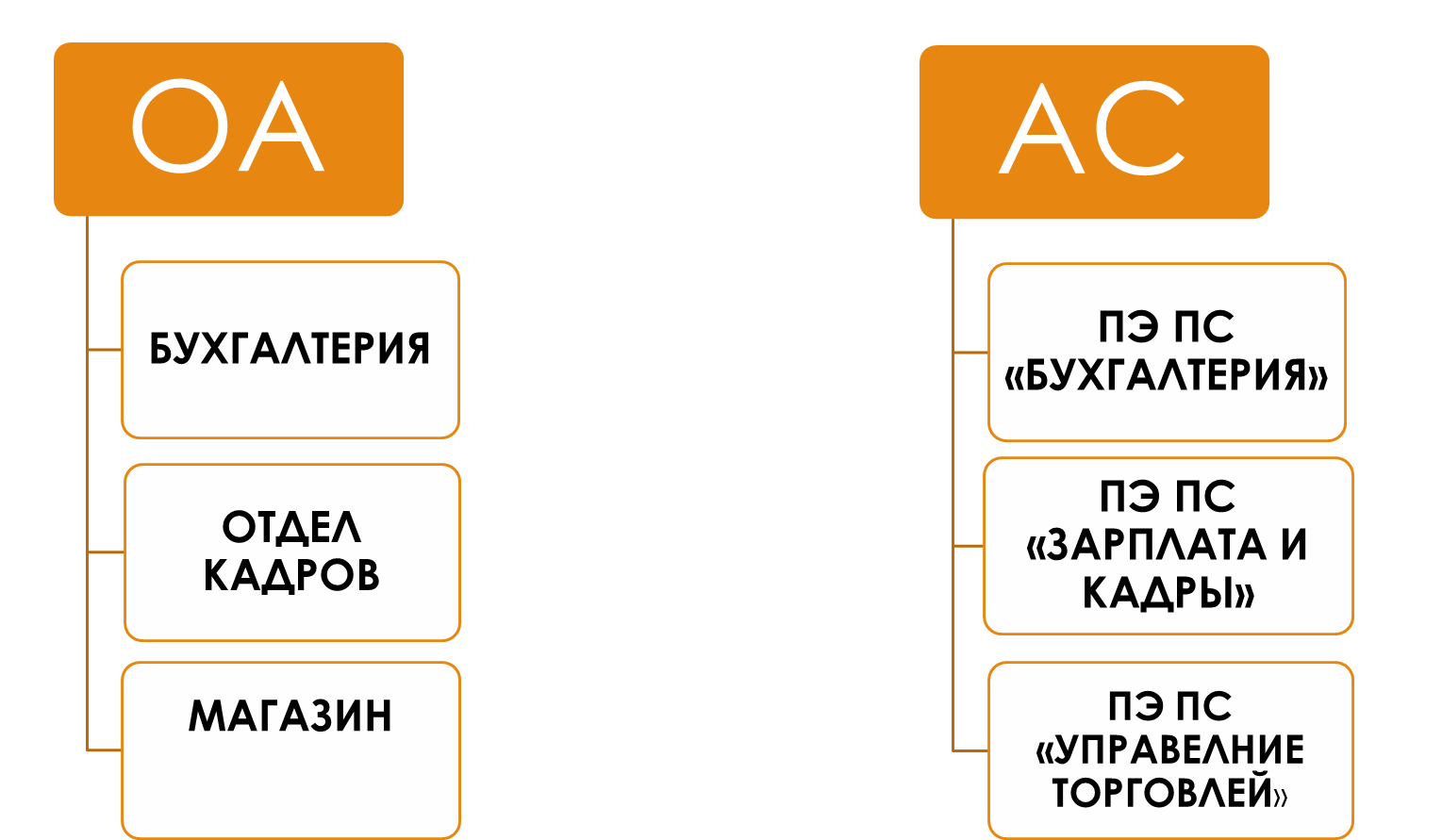
Функциональная структура в методология ARIS представляется через нотации VAD или Process Landscape. Данные нотации позволяют представлять функциональную структуру предприятия через иерархию и через последовательность (цепочку) процессов.

Элемент «Процесс». Основной элемент диаграмм Process Landscape

# 

# **Вопрос 14. Влияние функциональной структуры объекта автоматизации на структуру автоматизированной системы**

Организационная структура объекта автоматизации определяет структуру автоматизированной системы. Функциональная структура определяет состав ее программных элементов: алгоритмов, процедур, интерфейсов и т.д.



**Программное обеспечение АС**

# **Вопрос 15. Информационная структура и информационная модель объекта автоматизации**

**Информационная структура объекта автоматизации** – это совокупность документов, архивов, правил и норм ведения документооборота, используемые на объекте автоматизации. В ИС ОА входят так же средства автоматизации (автоматизированные и информационные системы, базы и банки данных, электронные архивы и т.д.), используемые объектом. Все элементы ИC ОА относятся либо к внемашинному (бумажному) либо внутримашинному (цифровому) обеспечению.

**Информационная модель объекта автоматизации.**

Под информационной моделью объекта понимается набор некоторых параметров, которые содержат необходимую информацию об объекте, системе объектов, процессе или явлении.

Целью создания информационной модели объекта является обработка данных об объекте с учетом его внутренних и внешних связей. Для того чтобы такую обработку можно было автоматизировать, для рассматриваемой модели составляют формализованное описание, доступное компьютерной обработке.

ИМ можно описывать различными способами: табличным, графическим, текстовым и т.д.

В методологии ARIS информационную модель можно выполнить с помощью нотации General diagram.

Основным носителями данных и информации в организациях и компаниях являются документы, являющиеся обязательными элементами ИМ. Поэтому ИМ можно представлять через информационные связи между документами.

Также документы, участвующие в ИМ, описываются через словарь данных, в котором указываются все атрибуты документа, их назначение, типы данных и источники данных.

Каждый документ в ИМ представляется через макет данных – графическую форму документа.

# 

# **Вопрос 16. Документ и его структура**

Документ – это информационное сообщение в бумажной, звуковой или электронной форме, оформленной по определенным правилам, заверенное в установленном порядке и имеющем юридическую силу. Документ должен быть оформлен по заданным правила и иметь определенную структуру, элементарной единицей которой является реквизит документа. Отсутствующий обязательный реквизит лишает документ юридической значимости и делает ничтожной любую сделку и недостоверной любую информацию. То есть, документ может подтверждать правовые обязательства или деловую деятельность компании только в случае, если соблюден установленный порядок его оформления.

Реквизиты документа — это перечень обязательных сведений, которые должны быть представлены в зависимости от вида документа. При этом важен не только состав и содержание того или иного реквизита, но и то, как он оформлен, и где расположен.

Перечень реквизитов у всех документов свой, в то же время можно выделить общую часть реквизитов, которые встречаются в документах достаточно часто.

**Номер документа** – реквизит документаоднозначно позволяющей сослаться на документ. Структура номера определяется в каждой организации самостоятельно.

**Дата документа –** реквизит, указывающийна дату его подписания, утверждения или дату событий, зафиксированного в документе.

**Наименование документа -** реквизит являющийся одним из важнейших, поскольку позволяет судить о назначении документа, определяет состав реквизитов и структуру текста. Наименование документов проставляется на всех документах.

**Наименование организации – составителя -** реквизит называет автора документа, которым может быть организация, структурное подразделение, должностное или физическое лицо.

**Визы согласования.** Указываются данные согласующих документ сторон.

**Заголовок к тексту.** Реквизит является обязательным для всех документов, кроме документов, составленных на бланке формата А5, и необходим для регистрации и поиска документа.

**Текст документа.** Текст документа составляют на русском или национальном языке в соответствии с законодательством Российском Федерации и субъектов Российской Федерации о государственных языках. Тексты документов оформляются в виде связного текста, анкеты, таблицы или в виде соединения этих структур.

**Подпись.** Подпись – обязательный реквизит, обеспечивающий удостоверение документа и придающий ему юридическую силу.

**Ответственный исполнитель документа.** Указывает сотрудника, которому поручено исполнение данного документа или который разработал данный документ (для исходящих и внутренних). Исполнитель документа всегда один и только один. Иногда встречаются два исполнителя документа (для документов длительного исполнения при смене кадрового состава предприятия).

**Документ-основание –** реквизит, указывающий на документ, породивший данный.

Каждый документ состоит из трех частей:

**Заголовочная часть** содержит следующие характеристики документа и учитываемого объекта: наименование учитываемого объекта (предприятия, организации, работающего); характеристики документа (индекс, код по общегосударственному классификатору управленческой документации - ОКУД); наименование документа; зона для представления кодов постоянных для документа реквизитов-признаков. В заголовочной части отражается в основном текстовая информация, которую необходимо закодировать для автоматизированной обработки.

**Содержательная часть** строится в виде таблицы, состоящей из строк и граф, где располагаются количественно-суммовые основания и их названия, которые обычно размещены в левой части таблицы. Документы, как правило, являются многострочными, с постоянным или переменным составом подлежащего таблицы. Все производные строки и графы документа имеют подсказки.

**Оформляющая часть** содержит подписи юридических лиц, отвечающих за правильность его составления, а также дату заполнения документа

# **Вопрос 17. Учет документооборота в организации. Жизненный цикл документа.**

Для исключения утери документов и их неисполнения в организациях ведется регистрация документооборота.

Регистрация документа — запись учетных данных о документе по установленной форме, фиксирующей факт его создания, отправления или получения путем внесения его в регистрационную форму с присвоением учетного регистрационного номера и записью в форму основных сведений о документе, что позволяет создать базу данных о документах учреждения для последующего контроля за сроками исполнения документов и справочной работы по ней.

Существуют три формы регистрации документов: журнальная; карточная; автоматизированная.

**Журнальная форма**. При большом количестве полученных документов процесс регистрации занимает много времени и задерживает дальнейшую работу с документами. Журнальная система регистрации часто приводит к повторной регистрации документов в других структурных подразделениях, куда документ передается. Но самый главный недостаток журнальной системы регистрации — это невозможность организовать контроль за сроками исполнения документов и оперативно вести информационно-справочную работу по документам.

**Карточная форма**.Карточная система регистрации позволяет преодолеть эти недостатки, так как карточки можно располагать в картотеках в любой последовательности, да и регистрировать документы могут несколько человек одновременно, создавая под копирку нужное количество экземпляров регистрационной карточки. Форма регистрационной карточки может быть определена самим учреждением и записывается в инструкции по делопроизводству.

Любой документ вне зависимости от его структуры или содержания проходит ряд стадий, которые в целом называются жизненным циклом документа.

Жизненный цикл документа - тип поведения документа от момента формирования до момента передачи в архив (на хранение) или уничтожения.

Все документы проходят через четыре основных этапа жизненного цикла (некоторые этапы могут повторяться, а некоторые имеют место только один раз):

* документ создаётся и утверждается;
* документ выполняет свою основную функцию и попадают в архив;
* документ при необходимости извлекаются из архива, а затем снова архивируются;
* документ уничтожается.

**Вопрос 18.** Виды документов, используемые в автоматизированных системах

**Справочные документы** предназначены для хранения постоянной или условно-постоянной информации ОА. К этим документам относятся различного рода справочники, нормы, тарифы, расписания, маршруты и т.д.

**Оперативные документы** предназначены для фиксации про­цессов, событий, которые проходят в процессе функционирования ОА.

**Отчетные документы** предназначены для представления резуль­татов обработки информации, которые содержатся в справочных и/или оперативных документах ОА.

# 

# **Вопрос 19. Бизнес-процесс и его элементы**

Любая современная компания является сложной организационной структурой с большим объемом документов, циркулирующих как внутри компании, так и между компанией и другими физическими и юридическими лицами. Процессный подход позволяет рассматривать деятельность учреждения как связанную систему деловых процессов, каждый из которых протекает во взаимосвязи с другими процессами или внешней средой. В основе процессного подхода лежит понятие бизнес-процесса.

**Бизнес-процесс**— это совокупность взаимосвязанных мероприятий или работ, выполняемых по определенным технологиям, преобразующим входной поток бизнес-процесса в выходной поток и направленным на создание определённого ценного продукта или услуги для потребителей.

Изучение бизнес-процесса позволяет целостно описать ОА как сложную систему т.к. в нем представлены все составляющие ОА, их функции и взаимосвязи между ними.

Вход бизнес-процесса –продукт, который в ходе выполнения бизнес-процесса преобразуется в выход. Входы процесса поступают в процесс извне (сырье, материалы, полуфабрикаты, документация, информация, персонал, услуги и т.д.)

Выход бизнес-процесса – результат (продукт, услуга) выполнения бизнес-процесса (готовая продукция, документация, информация, персонал, услуги и т.д.)

Ресурс бизнес-процесса – материальный или информационный объект, постоянно используемый для выполнения процесса, но не являющийся входом процесса (инструменты, оборудование, информационные ресурсы, персонал и т.д.)

Управление бизнес-процесса – ответственное лицо (владелец бизнес-процесса), которое отвечает за слаженное функционирование бизнес-процесса, управляет ходом бизнес-процесса и несет ответственность за результаты бизнес-процесса, а также методики, инструкции, правила, законодательство, согласно которым протекает бизнес-процесс.

# 

# **Вопрос 20. Проектирование процессной модели ОА**. **Нотация eEPC (Extended Events Process Chain) для проектирования процессной модели**

Процессная модель - описание деятельности предприятия в виде совокупности бизнес-процессов, непрерывных взаимосвязанных функций. Процессное моделирование осуществляется с помощью графических нотаций, текстового описания процессов и создания структуры процессов (например, дерева бизнес-процессов).

Процессная модель описывается в виде последовательности функций, реализуемой отдельными элементами организационной структуры. Модель бизнес-процесса связана с информационными, материальными и/или финансовыми потоками между функциями. Существует ряд нотаций бизнес-процесса, который применяется при моделировании бизнес-процессов. Нотации, как правило, отражают концептуальный подход (и методологию) разработчиков нотации к описанию и моделированию бизнес-процессов.

К числу наиболее практически важных, относится основная нотация архитектуры ARIS — нотация eEPC, что означает «расширенная цепочка процесса, управляемого событиями».

Данная методология проектирования рассматривает бизнес-процесс как последовательность событий и действий. Каждый процесс начинается с некоторого события и порождает некоторое действие, которое также порождает некоторое событие. Цепочка бизнес-процесс должна начинаться и заканчиваться событием.

При построении модели в ARIS eEPC должны неукоснительно соблюдаться следующие правила:

* каждая функция инициируется и завершается событием;
* в каждую функцию не может входить более одной стрелки, запускающей выполнение функции, и выходить более одной стрелки, описывающей завершение выполнения функции.

**Event**- состояние, которое является существенным для целей управления бизнесом и оказывает влияние или контролирует дальнейшее развитие одного или более бизнес-процессов. Элемент отображает события, активизирующие функции или порождаемые функциями. Внутри блока помещается наименование события. Событие именуется отглагольным существительным.

**Activity** - действие или набор действий, выполняемых над исходным объектом с целью получения заданного результата. Внутри блока помещается наименование функции (глагол или отглагольное существительное). Временная последовательность выполнения функций задается расположением функций на диаграмме процесса сверху вниз. Функция именуется глаголом или отглагольным существительным.

**Entity**(cущность) – смысловая единица даталогической модели (ER-model, модель «сущность - связь»).

**Database** – источник данных.

**Document** – бумажный или электронный носитель информации.

**IT system –** информационная система.

**Product –** ресурс или услуга, используемый как для входа в функцию, так и как результат ее выполнения.

**Risk-** возможная опасность того, что процесс не достигнет цели.

**Process interface –** элемент, обозначающий внешний (по отношению к текущей диаграмме) процесс или функцию. Используется для указания взаимосвязи процессов. Обозначает предыдущий или следующий процесс по отношению к диаграмме рассматриваемого процесса; обозначает процесс, откуда поступил или куда передается объект. Внутри блока помещается наименование внешнего процесса.

**AND (И) –** логический оператор, определяющий, что все события (или функции), объединенные данным оператором должны выполняться одновременно.

**XOR (Исключающее ИЛИ)-** логический оператор, определяющий, что должно выполняться только одно из событий (или функций), объединенных данным оператором.

**OR (ИЛИ) -** логический оператор, определяющий, что должно выполняться хотя бы одно из событий (или функций), объединенных данным оператором.

# **Вопрос 21. Технология проектирования баз данных**

Проектирование БД – это совокупность действий, направленных на последовательное преобразование требований заказчика к информационной системе и базе данных в частности в технический проект базы данных.

Исходной информацией для проектирования БД являются результаты процесса «Определения и анализа требований к БД», результаты обследования ОА (описания задач, документов, словарь данных и другие).

Результат проектирования базы данных – набор документации, описывающий проект структуры БД (концептуальная модель, логическая модель и физическая модель).

Проектирование БД – это итеративная процедура, которая включает реализацию трех процессов:

- Системно независимая модель «Сущность – связь»

- СУБД-ориентированная реляционная модель

- СУБД-ориентированная реализация объектов базы данных

Процесс проектирования концептуальной модели БД, заключающийся в создании системно независимой модели предметной области. Под предметной областью понимается множество всех объектов, свойства которых и отношения, между которыми рассматриваются в рамках некоторого исследования или в процессе деятельности. Модель предметной области – это некоторая система, адекватно имитирующая структуру или функционирование исследуемой предметной области.

Процесс проектирования логической модели БД, заключающийся в представлении концептуальной модели предметной области через совокупность реляционных отношений, находящихся в тех или иных связях друг с другом. Данная модель уже создается с ориентацией на ту или иную СУБД.

Процесс проектирования физической модели базы данных – это реализация логической модели базы данных через совокупность объектов базы данных, поддерживаемых выбранной СУБД.

Каждый из процессов соответствует определенному уровню детализации проектирования структуры БД и на каждом уровне используются свои модели данных:

В рамках процесса «Проектирование концептуальной модели» используется модель типа «сущность-связь».

В рамках процесса «Проектирование логической модели» может использоваться реляционная модель данных.

В рамках процесса «Проектирование физической модели» используются модели, которые поддерживаются выбранной для реализации БД СУБД.

# 

# **Вопрос 22. Концептуальное проектирование базы данных. Модель «Сущность - Связь»**

В основе проектирования ИС лежит моделирование предметной области. Для того чтобы получить адекватный предметной области проект ИС в виде системы правильно работающих программ, необходимо иметь целостное, системное представление модели, которое отражает все аспекты функционирования будущей информационной системы. При этом под моделью предметной области понимается некоторая система, имитирующая структуру или функционирование исследуемой предметной области и отвечающая основному требованию – быть адекватной этой области.

Предметная область - множество всех объектов, свойства которых и отношения, между которыми рассматриваются в рамках некоторого исследования или в процессе деятельности

Модель предметной области – это некоторая система, адекватно имитирующая структуру или функционирование исследуемой предметной области

К моделям предметной области предъявляются следующие требования:

* формализация, обеспечивающая однозначное описание структуры предметной области;
* понятность для заказчиков и разработчиков на основе применения графических средств отображения модели;
* реализуемость, т.е. наличие средств физической реализации модели предметной области в ИС;
* обеспечение оценки эффективности реализации модели предметной области на основе определенных методов и вычисляемых показателей

При моделировании предметной области используются следующие подходы:

* Структурно-функциональный
* Объектно-ориентированный
* Процессный

Концептуальная модель — это отражение смысловой структуры предметной области через ее значимые семантические единицы и связи между ними.

Основным представлением на данный момент КМ является модель «сущность-связь», основанная на диаграммной технике.

# 

# **Вопрос 22. Основные понятия методологии «Сущность - Связь»**

ER-модель – это модель предметной области, позволяющая описывать смысловую структуру предметной области. ER-модель используется при платформенно независимом, высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые смысловые единицы предметной области и обозначить связи, которые могут устанавливаться между ними.

Во время проектирования баз данных происходит преобразование схемы, построенной на основе ER-модели, в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

Для визуализации концептуальной модели, представленной в терминах модели «сущность-связь» используют различные графические нотации: нотация Чена, Crow’s Foot, IDF1X. Каждая из нотаций представляет из себя систему символики, с помощью которой представляется концептуальная модель предметной области в терминах модели «сущность - связь».

Основными понятиями модели «Сущность – Связь» являются: сущность, атрибут, связь, ключ.

Сущность – смысловая единица предметной области, имеющая в конкретном контексте независимое существование. Сущность- результат обобщения множества однородных объектов с одинаковыми свойствами. Имеет семантически значимое имя, выражаемое существительным: «Студент», «Преподаватель», «Сотрудник», «Отдел» и т. д.

Атрибут – это свойство сущности или связи, имеющее семантически значимое имя в форме существительного.

Связь – это отношение между сущностями, которое предполагает зависимость между атрибутами сущностей. Называется, как правило глаголом.

Экземпляр сущности – конкретные объекты, относящиеся к определенному типу сущности: конкретные персоны студентов, преподавателей, сотрудников, конкретные отделы и т. д. Экземпляры сущностей должны быть уникальными в контексте предметной области, т.е. экземпляр сущности должен идентифицироваться уникальным образом.

Ключ – уникальный идентификатор экземпляра сущности

Экземпляр связи – это отношение между экземплярами сущностей

# 

# **Вопрос 23. Систематизация предметной области**

Абстракция — отвлечение, обособление от тех или иных сторон, свойств или связей предметов и явлений для выделения существенных их признаков, которые с достаточной точностью представляют его в данной системе. Основная идея состоит в том, чтобы представить объект(явление) через минимальный набор характеристик при этом с достаточной точностью для решаемой задачи

Один из методов идентификации сущностей состоит **в изучении описания задач**, в которых определяются **конкретные функции** пользователя в процессе решения этих задач. Из этих описаний следует **извлечь все используемые в них существительные** или **сочетания существительного и прилагательного** (например, "табельный номер преподавателя", "фамилия сотрудника", "номер дисциплины", "адрес студента", "стоимость обучения" и другие). Затем среди них выбираются самые крупные объекты (люди, города) или представляющие интерес концепции и исключаются все существительные, которые просто определяют другие объекты. Например, свойства "Табельный номер преподавателя" и "фамилия преподавателя" могут быть объединены в сводном объекте под названием "преподаватель".

Альтернативный способ идентификации сущностей состоит в поиске объектов, которые существуют независимо от других. Например, объект "преподаватель" безусловно является сущностью, потому что любой сотрудник ВУЗа существует независимо от того, знаем мы его имя, адрес и номер телефона или нет

В процессе построения концептуальной модели, систематизация данных предметной области осуществляется на основе методов абстракции обобщения и абстракции агрегации, которыми люди интуитивно пользуются в обыденной жизни.

# 

# **Вопрос 24. Абстракция обобщения. Абстракция агрегации.**

Абстракция — отвлечение, обособление от тех или иных сторон, свойств или связей предметов и явлений для выделения существенных их признаков, которые с достаточной точностью представляют его в данной системе. Основная идея состоит в том, чтобы представить объект(явление) через минимальный набор характеристик при этом с достаточной точностью для решаемой задачи.

В процессе построения концептуальной модели, систематизация данных предметной области осуществляется на основе методов **абстракции обобщения** и **абстракции агрегации**, которыми люди интуитивно пользуются в обыденной жизни. Агрегация и обобщение используются для систематизации данных на этапе предпроектного анализа предметной области. Представляя разные контексты анализа одних и тех же элементов, взаимно дополняют друг друга. Если обобщение позволяет классифицировать объекты, то агрегация раскрывает структуру объектов.

**Абстракция обобщения** рассматривает набор различных подобных объектов (сущностей) как один поименованный обобщенный тип объекта.

Если набор однородных объектов можно обобщить до одного понятия, описывающего все элементы этого набора, то элементы рассматриваемого набора называются категориями, а обобщающее понятие– обобщением.

Абстракция обобщения имеет семантику «это»: «Математика», «Информатика» — это «Предмет» и т. д.

**Абстракция агрегации** – конструирование понятия или объекта из других базовых объектов. В абстракции агрегации связь между объектами рассматривается как новый объект. Абстракция агрегации имеет семантику «есть часть»: «Фамилия» есть часть «Сотрудник», «Адрес» есть часть «Сотрудник» и т. д.

Если набор различных неоднородных объектов образует в своей совокупности новое понятие (объект), то элементы такого набора называются компонентами (атрибутами), а понятие (объект) – отношением (сущностью). Говорят, что отношение агрегирует компоненты или компоненты агрегируются отношением.

# 

# **Вопрос 25. Проектирование концептуальной модели**

Построение КМ возможно путем использования следующих подходов:

«От данных» – КМ создается путем преимущественно метода агрегирования на основе составленного словаря данных.

«От задач» – КМ создается как совокупность ЛКМ, каждая из которых отражает отдельную функциональную задачу ОА.

«От документов» - КМ создается как совокупность ЛКМ на основе документооборота ОА.

Методика построения КМ представляется в виде трехэтапной процедуры:

**Этап 1**. «Проектирование локальных КМ для ОА».

В качестве исходных данных для этого подхода «От задач» удобно использовать процессную модель, выполненную в нотации eEPC. Данная нотация представляет процессы, происходящие в рамках ОА в виде последовательности «Событие - Действие». Действия выступают как функциональные задачи, каждая из которых ложится в основу отдельной ЛКМ.

Если выбран подход «От документов», то для каждого отдельного справочного и/или оперативного документа ОА разрабатывается отдельная локальная КМ (ЛКМ). Вначале рекомендуется создание ЛКМ для справочных документов. Затем создаются ЛКМ для оперативных документов. Для отчетных документов ЛКМ не создаются, так как эти документы содержат производную информацию, которая содержится в справочных и/или оперативных документах.

**Этап 2.** «Объединение локальных концептуальных моделей» в единую (глобальную) КМ ОА. При объединении ЛКМ в единую КМ используются такие способы объединения как объединение идентичностей, агрегация и обобщение.

Идентичными являются два или более элементов, если они имеют одинаковые семантические значения. Для идентичных элементов не является обязательным иметь одинаковые синтаксические описания. Обычно требуется глубокое понимание ОА, чтобы установить существование идентичности элементов.

Агрегация выражает такую связь понятий, которая семантически обозначает как «часть-целое». Агрегация используется в тех случаях, когда необходимо синтезировать сложное понятие, состоящее из совокупности более простых понятий.

Обобщение – это такая форма порождения нового понятия на основе одного или нескольких подобных понятий, когда обобщающее понятие сохраняет общие признаки исходных понятий, но игнорирует их более тонкие различительные признаки.

Так в приведенном выше примере сущности «Кладовщик» и «Менеджер» можно обобщить в единую сущность «Сотрудник». Также объединение ЛКМ будет происходить по идентичностям «ОП «Поступление ТМЦ»», «ТМЦ» - «ТОВАР», «Ед. хранения» - «Ед. измерения»

**Этап 3.** «Документирование КМ ОА»

# 

# **Вопрос 26. Логическое моделирование базы данных.**

Логическое моделирование – детализация и оптимизация концептуальной схему БД с учетом принимаемой модели БД (реляционной, иерархической, сетевой).

Логическая модель базы данных строится на базе концептуальной модели, являясь по сути ее детализацией до уровня описания атрибутов, доменов, механизмов целостности данных и т.д. Если концептуальная модель является платформенно независимой, то для построения логической модели уже требуется выбор конкретной СУБД, в терминах которой будет осуществляться построение логической модели.

Сам процесс проектирования логической модели состоит из трех этапов:

1. Построение ОТНОШЕНИЙ ЛМ на основе сущностей КМ
2. Нормализация ЛМ до требуемой нормальной формы (1НФ,2НФ,3НФ, НФБК,4НФ,5НФ)
3. Определение механизмов целостности данных

Логическое моделирование можно осуществлять с использованием CASE-средств (инструменты программной инженерии для проектирования программного обеспечения). Одним из CASE-средств логического и физического моделирования является пакет ERWIN, позволяющий автоматизировать построение логической и физической модели.

# 

# **Вопрос 27. Основные понятия логической модели**

**Декартово произведение множеств, отношение, атрибут, кортеж**

Пусть существует ***n*** множеств значений (типов данных или доменов) *R1, R2, …Rn*. Совокупность всех уникальных упорядоченных комбинаций (*r1,r2,…rn*), состоящих из ***n*** элементов, каждый элемент при этом принадлежит только одному из множеств , называется ***декартовым произведением***. Комбинация (*r1,r2,…rn*) называется ***кортежем***, а элементы *r1,r2,…rn*  - ***атрибутами***.

Например, есть множества ДОЛЖНОСТЬ, ФИО и ОКЛАД. Декартовым произведением этих множество будет являться множество всех уникальных кортежей, имеющих структуру (фио, должность, оклад), где атрибут кортежа фио взят из множества ФИО, атрибут должность из множества ДОЛЖНОСТЬ, атрибут оклад из множества ОКЛАД. ***Отношением***, построенном на данном декартовом произведении, будет являться множество тех кортежей, для которых присутствуют соответствия в предметной области.

**Свойства отношений, первичный ключ**

* Отсутствие кортежей-дубликатов

Достигается за счет введения в отношение атрибута (совокупности атрибутов), который будет иметь только уникальные значения, называемого ***ключом*** отношения. У отношения может быть несколько ключей. В таком случае они называются ***потенциальными ключами***. Если среди потенциальных ключей выбран один «основной», то он называется ***первичным ключом***. Остальные ключи переходят в разряд ***альтернативных***.

* Отсутствие упорядоченности кортежей

В таком случае должен быть механизм обращения к любому кортежу и этим механизмом является первичный ключ.

* Отсутствие упорядоченности атрибутов

Должен быть механизм обращения к любому атрибуту отношения. Этим механизмом является имя атрибута, которое должно быть уникальным в пределах отношения.

**Тип связи, экземпляр связи**

Если два типа сущности, находятся в смысловой связи друг с другом, то декартово произведение экземпляров этих сущностей будет являться ***типом связи*** между этими сущностями. А каждый элемент этого декартова произведения, представляя конкатенацию кортежей этих двух сущностей, будет являться ***экземпляром связи*** между этими сущностями (рис. 26.2). На логическом уровне экземпляр связи – это конкатенация кортежей отношений.

**Ссылочная целостность, внешний ключ**

Допустимые экземпляры связей между кортежами отношений определяются правилами, называемыми ограничения целостности связей или ссылочной целостностью. Ссылочно целостная связь двух отношений строится на связи «сильной» или идентифицирующей, т.е. способной в данной связи существовать независимо, и «слабой» сущностей, существование экземпляра которой невозможно без существования экземпляра «сильного» экземпляра. Например, СТУДЕНТ – ОЦЕНКА СТУДЕНТА.

В таком случае в структуру «слабой» сущности вводится дополнительное поле, которое будет идентифицировать принадлежность конкретного экземпляра «слабой» сущности к конкретному экземпляру «сильной» сущности. Такое поле называется «внешним» ключом. Значения внешнего колюча могут повторять только значения соответствующего первичного ключа.

**Мощность связи** обозначает максимальное количество экземпляров одной сущности, связанных с одним экземпляром другой сущности. На рис 26.3 представлены примеры мощностей связи, выполненные в нотации IDF1X. Так одному студенту соответствует много полученных им оценок, но каждая полученная оценка имеет только одного «хозяина». Т.е. это связь «один ко многим» Один предмет может читаться многими преподавателями и один преподаватель может читать много предметов. Это связь «много ко многим». Один паспорт принадлежит только одному студенту и каждый студент имеет только один паспорт. Это связь «один к одному».

**Полнота связи**

Полнота – возможность существования в сущности экземпляра, не связанного ни с одним экземпляром другой сущности.

Связь бывает:

**Полная** - ни в одной из связанных сущностей не может быть экземпляра, не связанного хотя бы с одним экземпляром другой сущности. (Пример: паспорт - студент)

**Неполная** - в обеих сущностях могут существовать экземпляры, не связанные ни с одним экземпляром другой сущности. (Пример: преподаватель и предмет)

**Частичной полноты** - промежуточный вариант, в одной из сущностей не могут быть связанные экземпляры, а в другой – могут. (Пример: студент и оценка)

**Домен**

Домен – это подмножество значений некоторого типа данных имеющих определенный смысл и по отношению к атрибутам домены играют роль области определения.

Домен характеризуется следующими свойствами:

* имеет уникальное имя;
* определен на некотором простом (скалярном) типе данных или на другом домене;
* может иметь некоторое логическое условие, позволяющее описать подмножество данных, допустимых для данного домена;
* может быть задан перечислением множества допустимых элементов данных;
* несет определенную смысловую нагрузку.

Примеры доменов:

Домен «зарплата» – множество вещественных чисел, превышающих минимальный размер оплаты труда;

Домен «образование» – множество строк символов, заданное перечислением {начальное, среднее общее, среднее специальное, высшее};

Домен «фамилия» – множество строк символов не длиннее 120;

Домен «оценка» – множество целых чисел в интервале от 0 до 10;

Домен «пол» – множество данных логического типа {мужской, женский};

Домен «номер» – множество положительных целых чисел.

# 

# **Вопрос 28. Построение отношений на основе сущностей КМ.**

При переходе от концептуальной модели к логической происходит трансформация сущностей в отношения. Областью определения множества экземпляров сущности является подмножество декартового произведения – отношение, определенное на доменах, связанных с ее атрибутами. Каждый кортеж такого отношения представляет экземпляр сущности.

Построение отношений на основе сущностей строится на следующих правилах:

* Удаление связей N:M

Если в концептуальной схеме присутствуют связи N:M, то их следует устранить путем определения промежуточной сущности.

Связь N:M заменяется двумя связями типа 1:M, устанавливаемыми со вновь созданной сущностью.

* Удаление атрибутов-синонимов

В отношении существую атрибуты, имеющие синонимичное значение

Такие атрибуты выносятся в отдельное отношение, ссылочно связанное с преобразуемым отношением.

* Перепроверка связей 1:1

Отношения, состоящие в таких связях, описывают одну и ту же сущность и могут быть объединены в одно отношение.

Решение о таком объединении должно приниматься на основании информации о превалировании того либо иного вида SQL-операций к таблицам, построенным на этих отношениях. Так DML-команды быстрее выполняются на коротких кортежах. В случае преобладания DML –команд стоит оставить связь «один ко одному», т.к. она оставляет кортежи отношений короче. Команда же Select на большом количестве входных таблиц будет выполняться медленнее, отнимая больше ресурсов у СУБД. В таком случае целесообразнее уменьшить декомпозицию данных в БД, объединив отношения, находящиеся в связи «один к одному» в одно.

**Вопрос 29. Нормализация логической модели БД**

База данных может быть спроектирована так, что с течением времени в ней будут накапливаться избыточные данные, теряться нужные данные при удалении ненужных, возникать различные противоречия при модификации данных и т.д.

Аномалии данных – противоречия модификации и накопления данных. Избыточность данных – накопление в таблицах БД лишних данных.

* При изменении фамилии необходимо просматривать все кортежи (аномалия обновления)
* Для внесения оценки необходимо повторно вводить данные ФИО (избыточность данных)
* Невозможность хранения информации о предмете без информации о студенте (аномалия удаления)
* Удаление записи о студенте может привести к исчезновению информации о предмете

Наличие в отношении атрибутов синонимов приводит к разреженности данных или необходимости ввода фиктивных данных.

Для снижения вероятности возникновения избыточности или аномалии данных логическая модель базы данных подвергается процессу нормализации. Т.е. структуру отношений логической модели базы данных изменяют таким образом, чтобы она соответствовала требованиям системы нормальных форм.

Каждая форма – набор требований. Каждая следующая нормальная форма обладает требованиями предыдущей и включает свои усиливающие требования. Отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору требований.

**Первая нормальная форма (1НФ)**

Отношение находится в первой нормальной форме, если в нем нет повторяющихся строк и все его атрибуты атомарны, т.е. неделимы

**Вторая нормальная форма (2НФ)**

Отношение находится во второй нормальной форме, если оно находится в 1НФ и каждый неключевой атрибут зависит от всего множества атрибутов, входящих в ключ, а не от его части.

**Третья нормальная форма (3НФ)**

Отношение находится в третье нормальной форме, если оно находится во 2НФ и каждый неключевой его атрибут не зависит от других неключевых атрибутов данного отношения.

Сильно нормализованные отношения больше соответствуют предметной области, легче в разработке, для них быстрее выполняются операции модификации базы данных. Однако выполнение операций выборки данных происходит медленнее. Большинство баз данных нормализованы до 3НФ.

При проектировании отношений следует придерживаться того принципа, что каждое отношение должно хранить данные об одном виде объектов (сущности) или об одном виде событий.

**Вопрос 30. Определение механизмов целостности данных**

Целостность базы данных — соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам (возраст должен быть положительным; заработная плата не должна быть меньше прожиточного минимума; имя должно состоять из букв и т.д.). Механизмы целостности данных обеспечивают согласованность данных в базе данных. Современные CASE-средства для проектирования баз данных позволяют определять механизмы целостности данных на этапе проектирования логической модели.

К механизмам целостности данных относятся:

* Типы данных и домены
* Первичный и внешний ключ
* Ограничение на значения колонок CHECK
* Умолчания DEFAULT
* Автоинкремент IDENTITY
* NULL, NOT NULL, UNIQUIE
* Триггеры

**Типы данных и домены**

Тип данных – это множество значений и операций над ними. Стандарт ANSI SQL определяет следующие общие типы данных: строковый (символьный), числовой, логический, дата-время, интервальный. Внутри данных типов могут быть те или иные подтипы, поддерживаемые различными СУБД.

Домен –это подмножество значений некоторого типа данных имеющих определенный смысл (домен «оценка» – множество целых чисел в интервале от 0 до 10).

**Первичный и внешний ключ**

Первичный ключ - это атрибут (либо совокупность атрибутов), значения которого должны быть уникальными. Также первичный ключ не может принимать значение NULL. Данный механизм предназначен для идентификации кортежей в отношении. Внешний ключ – это атрибут, идентифицирующий принадлежность кортежей со «слабыми» (т.е. подчиненными, порожденными) данными «сильным» данным (т.е. определяющим, порождающим). Например, кортеж, содержащий персональные данные студента из отношения «Студент», будет выступать в сильной, идентифицирующей роли по отношению к кортежам, содержащим данные о полученных студентом оценках, из отношения «ОценкиСтудента». В кортеже из отношения «Студент» будет присутствовать атрибут, идентифицирующий кортеж, называемый первичным ключом. Принадлежность кортежей с оценками в отношении «ОценкиСтудента» кортежу из отношения «Студент» будет определяться по значениям внешнего ключа, повторяющим значения соответствующего первичного ключа. Таким образом, пара первичный ключ – внешний ключ является механизмом целостности данных, обеспечивающим ссылочную целостность.

**Ограничения на значения колонок**

Механизм целостности данных позволяет ограничить диапазон значений, принимаемых атрибутом в пределах назначенного ему типа данных. Например, можно ограничить значения, принимаемые атрибутом «Оценка», определенном на целочисленном типе данных INTEGER, диапазоном от 0 до 10: CHECK (KURS BETWEEN 0 AND 10). Таким образом, значения, принимаемые данным атрибутом будут соответствовать логике предметной области.

**Умолчания DEFAULT**

Данный механизм обеспечивает заполнение атрибута значением в любом случае: даже тогда, когда значение атрибута не вводится пользователем. Механизм DEFAULTвводит определенное в нем значение в атрибут, если иное не определено пользователем.

**Автоинкремент IDENTITY**

Механизм **IDENTITY** обеспечивает автогенерацию значений в атрибуте, увеличивая каждый раз следующее значение на заданное приращение. Также для автогенерации задается начальное значение. Механизм чаще всего используется для автогенерации значений первичного ключа.

**NULL, NOT NULL, UNIQUIE**

Механизм NULL определяет для атрибута возможность не содержать какое-либо значение. NOT NULL – определяет требование того, что в атрибуте не может содержаться пустых значений.

UNIQUE определяет требование того, что в атрибуте, не являющемся первичным ключом, должны быть уникальные значения. Также данный механизм в отличие от первичного ключа позволяет атрибуту принимать значение NULL.

**Триггеры**

Это специальная хранимая процедура, подчиненная таблице базы данных, автоматически запускающаяся при попытке применения к данной таблице команды DML, на которую триггер настроен. Триггеры позволяют осуществлять каскадные изменения в таблицах, тем самым поддерживая согласованность в них данных. Современные CASE-пакеты для разработки баз данных позволяют определить триггеры на этапе проектирования логической модели базы данных.

# 

# **Вопрос 31. Физическая модель базы данных**

Физическая модель базы данных — это описание состава и структуры объектов базы дынных и содержит все детали, необходимые СУБД для создания базы данных.

В современных СУБД ФМ БД базируется на понятии SCHEMA. SCHEMA – это объект, содержащий в себе описание набора объектов, из которых состоит база данных: таблицы, пользователи, индексы, механизмы целостности данных, хранимые процедуры, триггеры, представления и т.д. Исходными данными для создания ФМ является ЛМ, в которой уже должны присутствовать нормализованные отношения, должны отсутствовать связи «много ко многим», определены механизмы целостности данных. При переходе от ЛМ к ФМ отношения трансформируются в таблицы, атрибуты отношения в столбцы таблицы, кортежи отношения в строки. Связи между таблицами поддерживаются с помощью пары первичный ключ – внешний ключ. Создание ФМ представляет из себя генерацию скриптов для создания схемы базы данных на основе ЛМ. Это можно сделать как вручную, так и с помощью современных CASE-средств.

Создать схему ФМ БД можно с помощью следующих скриптов:

CREATE DATABASE *имя\_базы\_данных –* инструкция используется для создания самой базы данных. Как правило, в ней определяется помимо имени базы данных расположение и размер файлов БД, автоприращение БД, владелец БД и др. параметры.

CREATE TABLE *имя\_таблицы –* инструкция задает структуру таблицы: состав полей и типов данных, механизмы целостности данных: CHECK, NULL, NOT NULL, DEFAULT, IDENTITY, PK, FK. Также структуру таблицы можно переопределить с помощь команды ALTER TABLE *имя\_таблицы.*

CREATE STORED PROCEDURE *имя\_процедуры* – инструкция создает хранимую процедуру в базе данных.

CREATE VIEW *имя\_предсталвения* – инструкция создает именованный запрос в базе данных.

CREATE TRIGGER ON *имя\_таблицы* FOR *DML\_команда* AS *тело\_триггера –* инструкция создает триггер, привязанный к определенной таблице и чувствительный к определенной команде DML.

CREATE INDEX *имя\_индекса* ON *имя\_таблицы (индексируемый\_столбец)*

CREATE TYPE *имя\_типа данных* FROM *описание\_типа* – создание пользовательского типа данных или домена.