1. **БАЗЫ ДАННЫХ И ФАЙЛОВЫЕ СИСТЕМЫ**
   1. **Дайте определение термина «данные».**

**Данные** - это информация, представленная в определенном виде, позволяющем автоматизировать ее сбор, хранение и дальнейшую обработку человеком или информационным средством

* 1. **Перечислите и дайте краткую характеристику направлений использования вычислительной техники.**

**Первое направление** – применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную.

**Второе направление** – это использование средств вычислительной техники в автоматических или автоматизированных информационных системах.

* 1. **Что понимается под информационной системой?**

**Информационная система** - программно-аппаратный комплекс, функции которого состоят в надежном хранении информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса

* 1. **Что такой «файл»?**

**Файл** - именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные

* 1. **Приведите краткую характеристику файловых систем обработки данных.**

**Файловые системы** - Набор программ, которые выполняют для пользователей некоторые операции, например создание отчетов. Каждая программа определяет свои данные и управляет ими.

* 1. **Приведите сильные и слабые стороны файловой систем обработки данных.**

**Сильные:** Простота использования, Универсальность, Легкость в управлении файлами, Быстрый доступ к данным, Надежность.

**Слабые:** избыточность данных, проблемы непротиворечивости данных, ограниченное разделение данных, ограничения по доступности данных, сложности в управлении.

* 1. **Кратко охарактеризуйте историю развития баз данных.**

История развития баз данных уходит корнями в 1960-е гг. Центральным событием, давшим старт развитию технологии БД, принято считать полет на Луну. Североамериканская компания Rockwell заключила контракт с правительством США на участие в проекте Apollo. Построение космического корабля включает в себя сборку нескольких миллионов деталей, поэтому была создана система управления файлами, отслеживавшая информацию о каждой детали. Однако в ходе последующей проверки обнаружилась огромная избыточность. Выяснилось, что почти все данные повторяются в двух и более файлах.

Ситуация требовала улучшения и множество специалистов усердно работали над созданием чего-то более удобного в использовании. Столкнувшись с задачей координации заказов на миллионы деталей, компания Rockwell в сотрудничестве с IBM в 1968 году разработала автоматизированную систему заказов. Названная IMS (Information Management System, система управления информацией), она заложила основу концепции БД.

**Первый этап развития СУБД** связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation – DEC), разных моделях HP (фирмы Hewlett Packard).

**Второй этап - появление персональных компьютеров.**

**Третий этап – распределенные многопользовательские БД.**

**Четвертый этап – современные перспективы развития СУБД.**

1. **ФУНКЦИИ СУБД. ТИПОВАЯ ОРГАНИЗАЦИЯ СУБД.**
   1. **Дайте определение термина «базы данных».**

**Базы данных** - поименованная совокупность структурированных данных, относящихся к определённой предметной области

* 1. **Дайте определение термина «СУБД».**

**СУБД** - комплекс программных и языковых средств необходимых для создания баз данных, поддержки их в актуальном состоянии и организации поиска в них необходимой информации, таким образом, базы данных – совокупность таблиц

* 1. **Перечислите основные функции СУБД и кратко их поясните.**

- СУБД должна предоставлять пользователям возможность сохранять, изменять и обновлять данные в БД

- СУБД должна иметь доступный конечным пользователям каталог, в котором хранится описание элементов данных

- СУБД должна иметь механизм, который гарантирует выполнение либо всех операций обновления данной транзакции, либо ни одной из них

- СУБД должна иметь механизм, который гарантирует корректное обновление БД при параллельном выполнении операций обновления многими пользователями

- СУБД должна предоставлять средства восстановления БД на случай ее повреждения или разрушения

- СУБД должна иметь механизм, гарантирующий возможность доступа к БД только санкционированных пользователей

- СУБД должна обладать способностью к интеграции с коммуникационным программным обеспечением

- СУБД должна обладать инструментами контроля за тем, чтобы данные и их изменения соответствовали заданным правилам

- СУБД должна обладать инструментами поддержки независимости программ от фактической структуры БД

- СУБД должна предоставлять некоторый набор различных вспомогательных служб

- Непосредственное управление данными во внешней памяти

- Управление буферами оперативной памяти

- Управление транзакциями

- Журнализация

- Поддержка языков БД

* 1. **Что такое транзакция?**

**Транзакция** - последовательность операций над БД, рассматриваемых СУБД как единое целое

* 1. **Для чего используется журнализация?**

Для восстановления базы данных после сбоя.

* 1. **Перечислите, какие компоненты входят в состав типовой СУБД.**

- ядро СУБД

- компилятор языка БД

- подсистема поддержки времени выполнения

- набор утилит

* 1. **Кратко охарактеризуйте компоненты СУБД.**

− процессор запросов – преобразует запросы в последовательность низкоуровневых инструкций для контроллера базы данных;

− контроллер базы данных – принимает запросы, проверяет внешние и концептуальные схемы для определения тех концептуальных записей, которые необходимы для выполнения запроса, в него входят обычно следующие основные компоненты: контроль прав доступа, процессор команд, средства контроля целостности, оптимизатор запросов, контроллер транзакций, планировщик, контроллер восстановления, контроллер буферов (память-диск);

− контроллер файлов – манипулирует файлами БД

− препроцессор языка DML – преобразует внедренные в прикладные программы DML-операторы в вызовы стандартных функций базового языка

− компилятор языка DDL – преобразует DDL-команды в набор таблиц, содержащих метаданные

− контроллер словаря – управляет доступом к системному каталогу и обеспечивает работу с ним

* 1. **Перечислите преимущества СУБД по отношению к файловым системам.**

− контроль за избыточностью данных

− непротиворечивость данных

− больше полезной информации при том же объеме хранимых данных

− совместное использование данных

− поддержка целостности данных

− повышенная безопасность

− применение стандартов

− повышение эффективности с ростом масштабов системы

− возможность нахождения компромисса для противоречивых требований

− повышение доступности данных и их готовности к работе

− улучшение показателей производительности

− упрощение сопровождения системы за счет независимости от данных

− улучшенное управление параллельностью

− развитые службы резервного копирования и восстановления

* 1. **Перечислите недостатки СУБД.**

− сложность и размер

− стоимость СУБД и дополнительные затраты на аппаратное обеспечение

− затраты на преобразование

− производительность

− серьезные последствия при выходе системы из строя

1. **РАННИЕ ПОДХОДЫ К ОРГАНИЗАЦИИ БД**
   1. **Какие подходы относятся к дореляционным моделям данных?**

- иерархические системы;

- сетевые системы;

- системы, основанные на инвертированных списках.

* 1. **Укажите сильные и слабые стороны дореляционных моделей.**

**Сильные:** − развитые средства управления данными во внешней памяти на низком уровне;

− возможность построения вручную эффективных прикладных систем;

− возможность экономии памяти за счет разделения подобъектов

**Слабые:** − сложность использования конечным пользователем, поскольку фактически необходимы знания о физической организации;

− прикладные системы напрямую зависят от этой физической организации;

− логика работы ранних СУБД перегружена деталями организации доступа к БД.

* 1. **Приведите характеристику иерархической модели данных.**

**Иерархическая модель данных** – представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

* 1. **Перечислите и дайте определение основных понятий иерархической БД: поле, тип сегмента, сегмент, экземпляр сегмента.**

**Поле** – неделимый элемент данных

**Сегмент** - Поименованная совокупность полей

**Экземпляр сегмента** – это некоторая логическая запись, описывающая конкретный объект предметной области

**Тип сегмента** - структурная схема экземпляра, в которой указаны имена составляющих ее полей

* 1. **Что относится к достоинствам и недостаткам иерархической модели?**

**Достоинства**: - Простота модели.

- Использование отношений предок/потомок.

- Быстродействие.

**Недостатки**: - Операции манипулирования данными в иерархических системах ориентированы, прежде всего, на поиск информации сверху-вниз, т.е. по данному экземпляру сегмента-отца можно найти все экземпляры сегментов-сыновей. Обратный поиск затруднен, а часто и невозможен.

- Дублирование данных на логическом уровне.

- Отношение «многие-ко-многим» непосредственно не поддерживается, что и является основным недостатком иерархических моделей.

* 1. **Приведите характеристику сетевой модели данных.**

**Сетевая модель данных** - расширение иерархического, позволяющая описывать связи многие-ко-многим

* 1. **Перечислите и дайте определение основных понятий сетевой модели: элемент данных, агрегат данных, запись, набор записей.**

**Элемент данных** – наименьшая поименованная единица данных

**Агрегат данных** – поименованная совокупность элементов данных внутри записи, которую можно рассматривать как единую целую.

**Запись** – поименованная совокупность элементов данных и/или агрегатов, которая не входит в состав никакого другого агрегата.

**Набор записей** – поименованная двухуровневая иерархическая структура, связывающая запись-владельца и записи-членов.

* 1. **Что относится к достоинствам и недостаткам сетевой модели?**

**Достоинства:** − наличие успешных реализаций систем управления базами данных, обеспечивающих эту сетевую модель

− простота реализации часто встречающихся в реальном мире взаимосвязей «многие ко многим».

**Недостатки**: − Сложность

− Возможная потеря независимости данных при реорганизации базы данных.

* 1. **Приведите характеристику систем, основанных на инвертированных списках.**

**Инвертированный список** - это структура данных, в которой порядок элементов инвертирован относительно оригинального порядка. Системы, основанные на инвертированных списках, используют эту структуру для решения различных задач.

- Поиск по значению

- Вставка и удаление

И т.д.

* 1. **Перечислите особенности БД, организованных с помощью инвертированных списков.**

− строки таблиц упорядочены системой в некоторой физической последовательности;

− физическая упорядоченность строк всех таблиц может определяться и для всей БД;

− для каждой таблицы можно определить произвольное число ключей поиска, для которых строятся индексы.

* 1. **Что относится к достоинствам и недостаткам систем, основанных на инвертированных списках?**

**Достоинства:** − более быстрый поиск

− возможность хранения элементов данных со сложной структурой.

**Недостатки**: – отсутствие строгого математического аппарата

– отсутствие средств для описания ограничений целостности базы данных

– большая трудоемкость программирования запросов к базе данных.

1. **ОБЩИЕ ПОНЯТИЯ РЕЛЯЦИОННОГО ПОДХОДА К ОРГАНИЗАЦИИ БАЗЫ ДАННЫХ**
   1. **Дайте краткую характеристику реляционной модели данных.**

**Реляционная модель** – это возврат к плоским файлам.

* 1. **Поясните на примере основные понятия реляционных БД: тип данных, домен, атрибут, кортеж, первичный ключ и отношение.**

**Тип данных** – тоже, что и в программировании.

**Домен** – это множество элементов

**Атрибут** – некоторая характеристика объекта

**Кортеж** – это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения.

**Первичный ключ** – это особенное поле, в котором сохраняется уникальный идентификатор записи.

**Отношение** – это множество кортежей, соответствующих одной схеме отношения.

* 1. **Что такое ключ? Какие виды ключей различают в реляционной модели данных?**

**Ключ** - Атрибут, значение которого идентифицирует кортеж,

**Виды:** − Простой ключ

− Сложный (составной) ключ

− Суперключ

− Искусственный ключ

− Естественный ключ

* 1. **Перечислите и дайте характеристику фундаментальных свойство отношений.**

- Отсутствие кортежей-дубликатов

- Отсутствие упорядоченности кортежей

- Отсутствие упорядоченности атрибутов

- Атомарность значений атрибутов

* 1. **Поясните трактовку реляционной модели данных согласно Дейту.**

Согласно Дейту реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: **структурной части**, **манипуляционной части** и **целостной части**.

* 1. **В чем заключается суть структурной части реляционной модели данных?**

В **структурной части** модели фиксируется, что единственной структурой данных, используемой в реляционных БД, является нормализованное n-арное отношение.

* 1. **В чем заключается суть манипуляционной части реляционной модели?**

В **манипуляционной части** модели утверждаются два фундаментальных механизма манипулирования реляционными БД – реляционная алгебра и реляционное исчисление.

* 1. **Какие требования входят в целостную часть реляционной модели данных?**

Первое требование называется требованием **целостности сущностей**. Любой кортеж любого отношения должен быть отличим от любого другого кортежа этого отношения, т.е. другими словами, любое отношение должно обладать первичным ключом.

Второе требование называется требованием **целостности по ссылкам.** Для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным.

1. **БАЗИСНЫЕ СРЕДСТВА МАНИПУЛИРОВАНИЯ РЕЛЯЦИОННЫМИ ДАННЫМИ**
   1. **В чем заключается разница между реляционной алгеброй и реляционном исчислении?**

**Реляционная алгебра** основана на теории множеств, а **Реляционное исчисление** базируется на математической логике.

* 1. **Какой язык называется реляционно полным?**

Конкретный язык манипулирования реляционными БД называется **реляционно полным**, если любой запрос, выражаемый с помощью одного выражения реляционной алгебры или одной формулы реляционного исчисления, может быть выражен с помощью одного оператора этого языка.

* 1. **В чем недостаток реляционной алгебры Кодда?**

- Восемь перечисленных операций по охвату своих функций, с одной стороны, избыточны, т.к. минимально необходимый набор составляют пять операций: объединение, вычитание, произведение, проекция и выборка. Три другие операции (пересечение, соединение и деление) можно определить через пять минимально необходимых. Так, например, соединение – это проекция выборки произведения;

- Этих восьми операций недостаточно для построения реальной СУБД на принципах реляционной алгебры. Требуются расширения, включающие операции: переименования атрибутов, образования новых вычисляемых атрибутов, вычисления итоговых функций, построения сложных алгебраических выражений, присвоения, сравнения и т.д.

* 1. **Какие операции реляционной алгебры Вы знаете?**

объединение, разность (вычитание), пересечение, декартово (прямое) произведение (или произведение), выборка (селекция, ограничение), проекция, деление и соединение

* 1. **Расскажите о базовых теоретико-множественных операциях реляционной алгебры Кодда.**

объединение, разность, пересечение и произведение

* 1. **Расскажите о специальных реляционных операциях Кодда.**

проекция, селекция, деление и соединение

* 1. **Расскажите о реляционном исчислении.**

Преимуществом реляционного исчисления перед реляционной алгеброй можно считать то, то пользователю не требуется самому строить алгоритм выполнения запроса. Программа СУБД сама строит эффективный алгоритм.

* 1. **Что такое исчисление кортежей и исчисление доменов?**

В **исчислении кортежей** областями определения переменных являются отношения базы данных, т.е. допустимым значением каждой переменной является кортеж некоторого отношения.

В **исчислении доменов** областями определения переменных являются домены, на которых определены атрибуты отношений базы данных, т.е. допустимым значением каждой переменной является значение некоторого домена.

1. **ПРОЦЕСС ПРОЕКТИРОВАНИЯ БД**
   1. **Какие области охватывает процесс проектирования БД?**

− проектирование объектов БД

− проектирование интерфейса взаимодействия с БД

− проектирование БД под конкретную вычислительную среду или информационную технологию

− проектирование баз данных под назначение

* 1. **Дайте характеристику трехуровневой архитектуры ANSI-SPARC.**

Архитектура **ANSI-SPARC** имеет три уровня: **внешний**, **концептуальный** и **внутренний.**

**Внешний уровень** – представление БД с точки зрения пользователей.

**Концептуальный уровень** – обобщающее представление БД.

**Внутренний уровень** – физическое представление базы данных в компьютере.

* 1. **Перечислите этапы проектирования БД.**

**Этап 1. Концептуальное проектирование** – создание схемы БД, включающего определение важнейших сущностей (таблиц) и связей между ними, но не зависящего от модели БД (иерархической, сетевой, реляционной и т. д.) и физической реализации (целевой СУБД).

**Этап 2. Логическое проектирование** – развитие концептуальной схемы БД с учетом принимаемой модели (иерархической, сетевой, реляционной и т.д.).

**Этап 3. Физическое проектирование** – развитие логической схемы БД с учетом выбранной целевой СУБД.

* 1. **Расскажите об этапе концептуального проектирования.**

Перед тем, как создавать информационную систему, предметная область должна быть описана – должна быть создана концептуальная модель предметной области.

* 1. **Какова цель этапа концептуального проектирования?**

**Целью** концептуального проектирования является создание концептуальной схемы данных на основе представлений о предметной области каждого отдельного типа пользователей.

* 1. **Что такое ER-модель? Кратко перечислите шаги по ее созданию.**

**ER-модель**— модель данных, позволяющая описывать концептуальные схемы предметной области.

**Шаги:** 1) Выделение сущностей

2) Определение атрибутов

3) Определение ключей

4) Определение связей

* 1. **Какие типы связей ER-модели Вы знаете?**

- один-к-одному

- один-ко-многим

- многие-ко многим

* 1. **Расскажите об этапе логического проектирования.**

На этапе логического проектирования необходимо провести нормализацию БД. Поэтому необходимо будет проверить концептуальную схему с помощью методов нормализации и контроля выполнения транзакций

* 1. **В чем заключается цель логического проектирования?**

**Цель логического проектирования** – развить концептуальную схему БД с учетом выбранной модели БД (иерархической, сетевой, реляционной и т.д.).

* 1. **Перечислите основные шаги этапа логического проектирования.**

1) Удаление и проверка элементов, не отвечающих принятой модели данных

* 1. Удаление связей N:M
  2. Удаление связей с атрибутами
  3. Удаление сложных связей (со степенью участия более 2)
  4. Удаление рекурсивных связей (со степенью участия 1).
  5. Удаление многозначных атрибутов (атрибутов имеющих несколько значений)
  6. Удаление избыточных связей
  7. Перепроверка связей 1:1

2) Проверка модели с помощью правил нормализации

3) Определение требований поддержки целостности данных

* 1. **Расскажите об этапе физического проектирования.**

Этап **физического проектирования** заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т.е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам физической» БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL).

* 1. **В чем заключается цель физического проектирования?**

**Цель физического проектирования** – преобразование логической схемы с учетом синтаксиса, семантики и возможностей выбранной целевой СУБД.

1. **ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БД НА ОСНОВЕ ПРИНЦИПОВ НОРМАЛИЗАЦИИ**
   1. **С каким проблемами можно столкнуться при проектировании БД?**

− Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было по возможности лучшим (эффективным, удобным и т.д.)? Часто эту проблему **называют проблемой логического проектирования** баз данных.

− Как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создание каких дополнительных структур (например, индексов) потребовать и т.д.? Эту проблему называют **проблемой физического проектирования** баз данных.

* 1. **Перечислите существующие виды аномалий.**

**аномалия обновления**: в одних кортежах будет новое значение оклада, а в других – старое.

**аномалия включения**: в отношении нельзя хранить данные о преподавателе, не ведущем в настоящее время никаких дисциплин, например, находящемся на повышении квалификации

**аномалии удаления**: удалив данные о предмете, можно удалить данные о преподавателе.

* 1. **Что такое нормализация?**

**Нормализация**— это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

* 1. **Сколько нормальных форм существует?**

− первая нормальная форма (1НФ);

− вторая нормальная форма (2 НФ);

− третья нормальная форма (3 НФ);

− нормальная форма Бойса-Кодда (БКНФ);

− четвертая нормальная форма (4 НФ);

− пятая нормальная форма, или нормальная форма проекции соединения (5 НФ).

Набольшую практическую значимость имеют **первые три нормальные формы.**

* 1. **Что такое функциональная зависимость?**

**Функциональные зависимости** являются отражением семантики взаимосвязи данных в предметной области. С каждым построенным отношением в БД связывается определенная совокупность функциональных зависимостей, которые являются в ряде случаев источником аномалий данных.

* 1. **Дайте определение первой нормальной формы.**

**Отношение находится в первой нормальной форме (1НФ**) - если все его атрибуты являются атомарными, т.е. состоящими из неделимых значений.

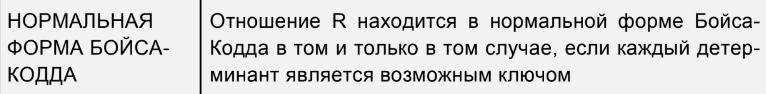
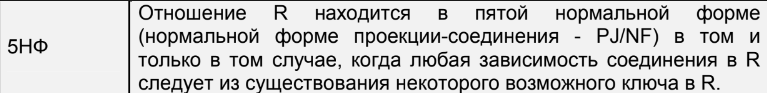
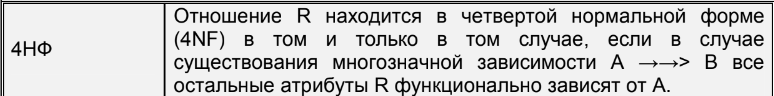
* 1. **Дайте определение второй нормальной формы.**

**Отношение находится во второй нормальной форме (2НФ)** -в том и только в том случае, когда находится в 1НФ, и каждый неключевой атрибут полностью зависит от первичного ключа.

* 1. **Дайте определение третьей нормальной формы.**

**Отношение R находится в третьей нормальной форме (3НФ)** - в том и только в том случае, если находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

* 1. **Кратко охарактеризуйте нормальные формы высшего порядка.**

****

1. **ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БД С ИСПОЛЬЗОВАНИЕМ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ: ER-ДИАГРАММЫ**
   1. **В чем заключаются недостатки процесс проектирования БД с помощью нормализации?**
2. Первоначальное размещение всех атрибутов в одном отношении является весьма неестественной операцией.
3. Невозможно сразу определить полный список атрибутов.
4. Несмотря на то, что процесс проектирования начинается с выделения некоторых существенных для приложения объектов предметной области («сущностей») и выявления связей между этими сущностями, реляционная модель данных не предлагает какого-либо механизма для разделения сущностей и связей.
5. Для проведения процедуры нормализации необходимо выделить зависимости атрибутов, что тоже очень нелегко, т.к. необходимо явно выписать все зависимости, даже те, которые являются очевидными.
   1. **Что такое семантическое моделирование?**

**Семантическое моделирование** - представляет собой моделирование структуры данных, опираясь на смысл этих данных.

* 1. **Перечислите основные компоненты семантической модели.**

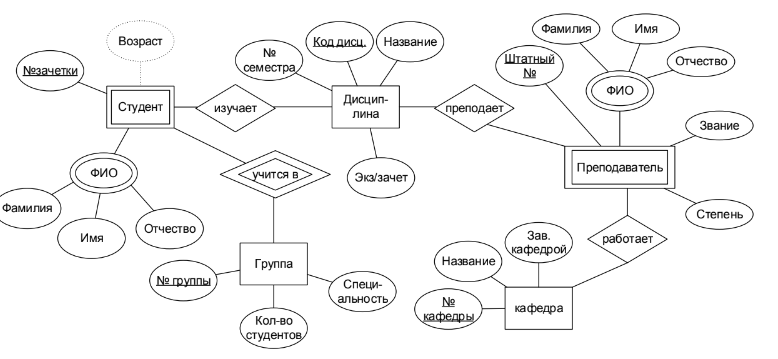
**− Сущность**

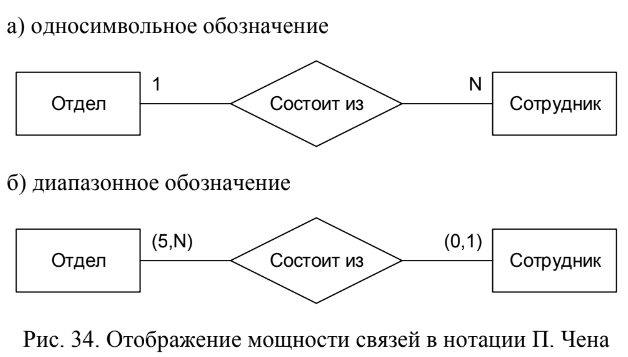
**− Атрибуты.** Каждая сущность состоит из набора атрибутов, которые описывают конкретные свойства сущности.

**− Связь** – описывает отношение между данными.

* 1. **Дайте характеристику ER-модели П Чена.**

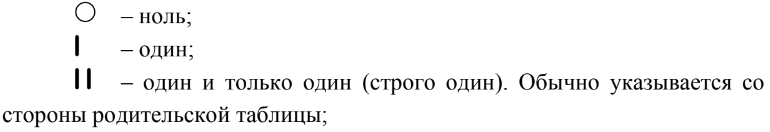
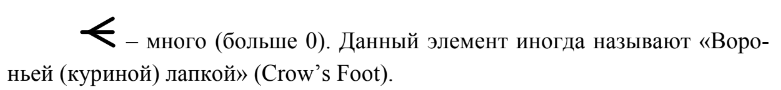
Независимая сущность отображается на диаграммах прямоугольником с одинарной рамкой, зависимая – с двойной. Атрибуты отображаются за пределами графического обозначения сущности в виде эллипсов, связанных одинарной линией с ним. Атрибуты, входящие в первичный ключ сущности, выделяются подчеркиванием имени. Эллипс многозначных атрибутов изображается с двойным контуром, производных – пунктирным. Если атрибут является составным, то атрибуты-компоненты отображаются в виде присоединенных к нему эллипсов. Связь между сущностями показывается в виде ромба с указанием имени связи внутри него. При этом, если ромб имеет двойную рамку, то связь – идентифицирующая, одинарную – неидентифицирующая. Вынесение атрибутов за пределы графического изображения сущности делает нотацию Чена похожей на семантические сети.

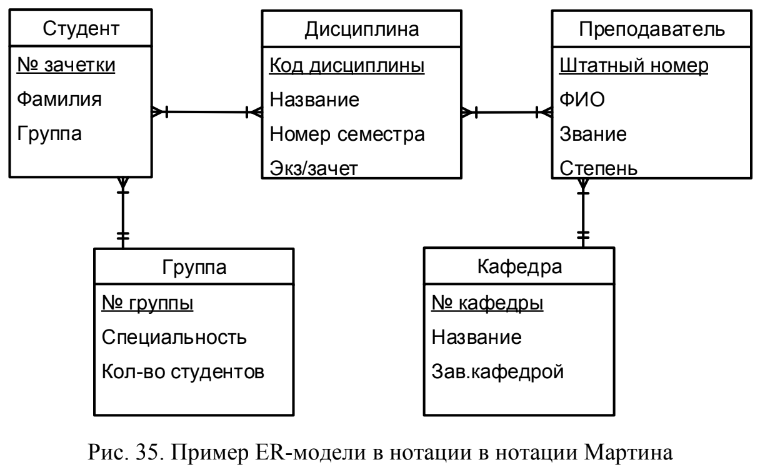




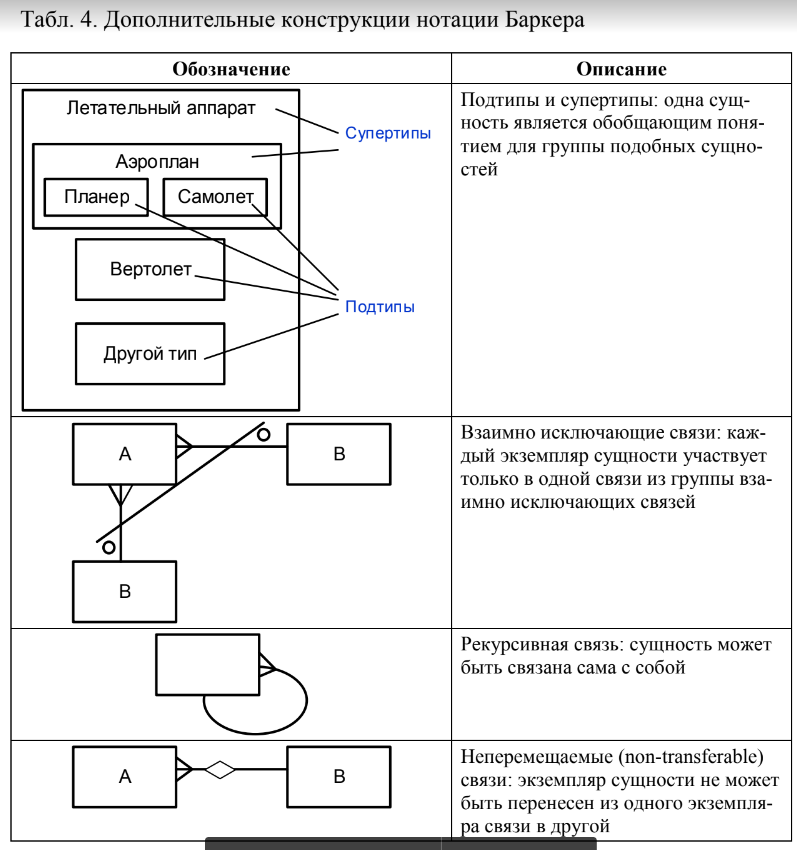
* 1. **Дайте характеристику ER-модели «воронья лапка».**

В нотации IE сущность отображается в виде прямоугольника, содержащего его имя. Согласно данной нотации, сущность изображается в виде прямоугольника, содержащем ее имя, выражаемое существительным. Имя сущности должно быть уникальным в рамках одной модели. При этом имя сущности – это имя типа, а не конкретного экземпляра данного типа. Экземпляром сущности называется конкретный представитель данной сущности. Атрибуты сущности записываются внутри прямоугольника, изображающего сущность, за счет этого диаграмма получается гораздо более компактной. Ключевые поля на диаграмме помещаются в верхнюю секцию прямоугольника (Рис. 35).

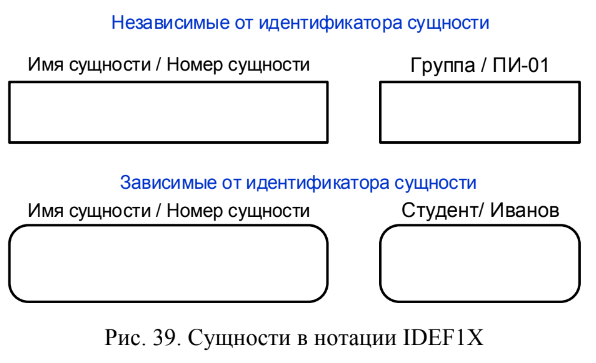
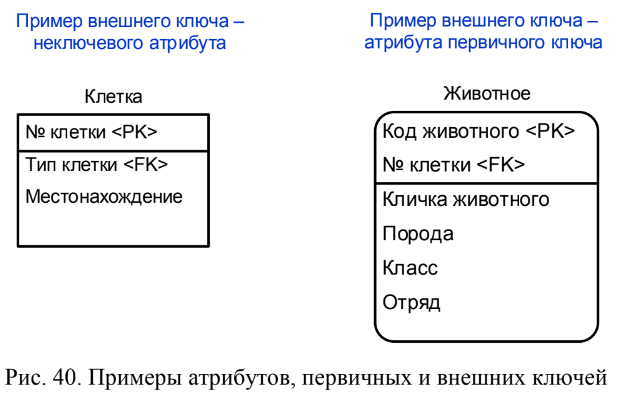
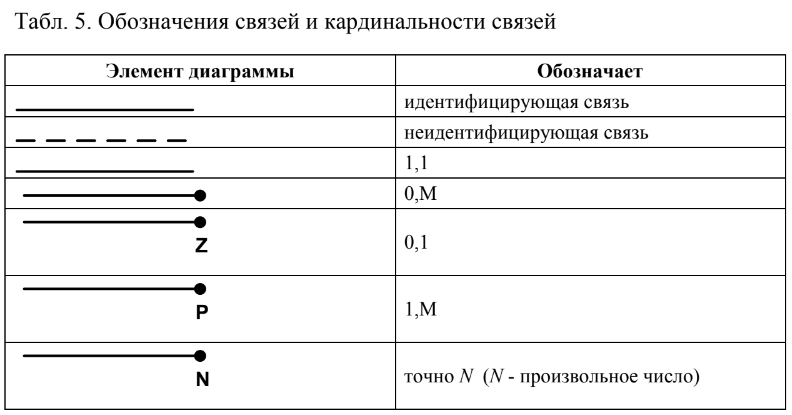
Мощности связей указываются не в виде буквенно-цифрового обозначений, а с помощью графических элементов:  

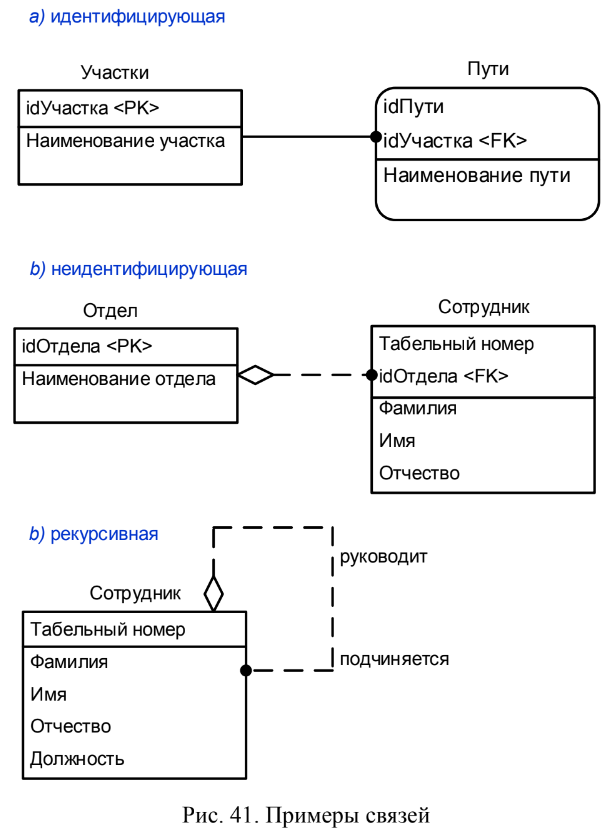
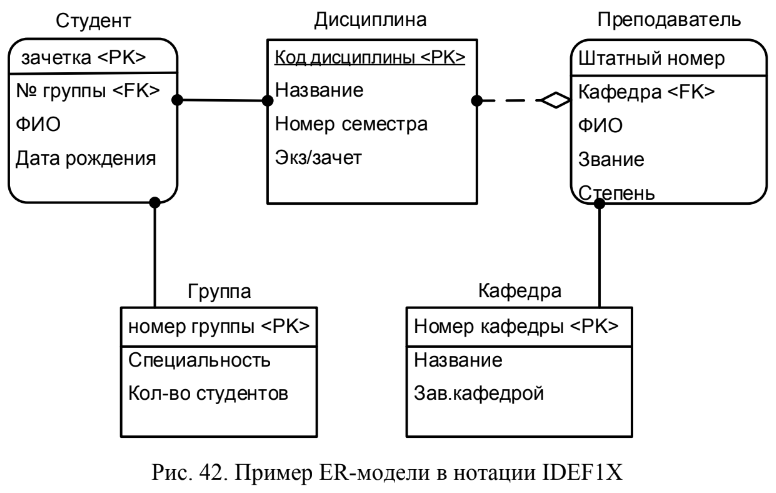


* 1. **Дайте характеристику нотации Баркера. **

****

* 1. **Дайте характеристику методологии IDEF1X.**

Обозначение сущности в графической нотации IDEF1X различается для зависимой и независимой сущности Рис. 39). Сущность в методологии IDEF1X является независимой (сильной, родительской, доминантной, владельцем), если сущность не зависит от существования другой сущности (другими словами, каждый экземпляр сущности может быть однозначно идентифицирован без определения его связей с другими сущностями, или уникальность экземпляра определяется только собственными атрибутами). Сущность называется зависимой (слабой, дочерней, подчиненной), если ее существование зависит от существования других сущностей. Терминология «родительская» – «дочерняя» и «владелец» – «подчиненный» также может использоваться в отношении двух зависимых сущностей, если экземпляры одной из них (дочерней, подчиненной) могут быть однозначно определены с использованием экземпляров другой (родительской, владельца), несмотря на то, что вторая сущность в свою очередь зависит от третьей сущности. Каждой сущности может быть присвоено уникальное имя и номер, разделяемые косой чертой « / » и помещаемые над блоком.   

* 1. **Перечислите шаги для получения реляционной схемы из ER-схемы.**

**Шаг 1.** Каждая простая сущность превращается в таблицу.

**Шаг 2.** Каждый атрибут становится возможным столбцом с тем же именем;

**Шаг 3.** Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы.

**Шаг 4.** Связи многие-к-одному (и один-к-одному) становятся внешними ключами.

**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:

**a)** все подтипы в одной таблице;

**b)** для каждого подтипа – отдельная таблица.

**Шаг 7.** Если в концептуальной модели имелись исключающие связи, то также возможны два способа:

- общий домен.

- явные внешние ключи.

1. **SQL (STRUCTURED ENGLISH QUERY LANGUAGE)**
   1. **Дайте краткую характеристику идентификаторов, констант, типов данных и ограничений в SQL.**

**Идентификаторы SQL** Идентификаторы в SQL используются для именования баз данных, таблиц, столбцов и других объектов в базе данных.

**Константы** представляют постоянные значения в SQL, которые не могут быть изменены в процессе выполнения запроса.

**Типы данных в SQL** определяют характеристики данных, которые могут быть сохранены в столбцах таблиц.

**Ограничение** - средство, позволяющее автоматически обеспечить целостность данных, определяемое при создании таблиц. Различают четыре типа ограничений: **ограничение первичного ключа** (Primary Key), **ограничение уникальности** (Unique), **ограничение внешнего ключа** (Foreign Key), **ограничение значением** (Check).

* 1. **Приведите общую характеристику языка SQL.**

**Язык SQL** предназначен для выполнения операций над таблицами (создание, удаление, изменения структуры) и над данными таблиц (выборка, изменение, добавление и удаление), а также некоторых сопутствующих операций.

* 1. **Какие части входят в структуру SQL?**

**Операторы SQL** можно условно разделить на два подъязыка: **язык определения данных** (Data Definition Language – DDL) и **язык манипулирования данными** (Data Manipulation Language – DML).

**DDL**: CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE VIEW, DROP VIEW, ALTER VIEW, CREATE INDEX, DROP INDEX.

**DML:** SELECT, UPDATE, INSERT, DELETE.

* 1. **Расскажите об операторе выбора SELECT.**

**SELECT –** ключевое слово, которое сообщает СУБД, что эта команда – запрос. Все запросы начинаются с этого слова с последующим пробелом. За ним может следовать способ выборки.

* 1. **Как в SQL обрабатываются агрегатные функции и группировка?**

Для применения **агрегатных функций** предполагается предварительная операция группировки. При группировке все множество кортежей отношения разбивается на группы, в которых объединяются кортежи, имеющие одинаковые значения атрибутов, которые заданы в списке группировки.

GROUP BY **группирует** записи данных и объединяет в одну запись все записи данных, которые содержат идентичные значения в указанном поле (или полях).

* 1. **Что такое вложенные запросы?**

В SQL вложенные запросы (или подзапросы) представляют собой запросы, вложенные внутрь других SQL-запросов. Они позволяют выполнять сложные запросы, используя результаты других запросов в качестве условий или данных для более обширных операций. Вложенные запросы можно использовать в различных частях SQL-запроса, таких как SELECT, FROM, WHERE, и т.д.

* 1. **Что такое внутренние и внешние объединения?**

**Внутреннее объединение (INNER JOIN)** возвращает записи из двух таблиц, если значение первичного ключа первой таблицы соответствует значению внешнего ключа второй таблицы, связанной с первой.

**Левое внешнее объединение:** в выборку включаются все записи таблицы, имя которой указано слева от оператора OUTER JOIN.

**Правое внешнее объединение:** в выборку включаются все записи таблицы, имя которой указано справа от оператора OUTER JOIN.

**Полное внешнее объединение:** в выборку включаются все записи из правой и левой таблиц.

* 1. **Как в SQL обрабатываются перекрестные запросы?**

**Перекрестный запрос** – способ группировки данных по двум измерениям, позволяющий отображать итоги в компактном результирующем наборе. В перекрестном запросе группировка выполняется по двум полям (группировка по строкам может быть больше, чем по одному полю), а итоговая функция применяется к полю, выводимому на пересечении строк и столбцов.

Структура перекрестного запроса следующая:

– в конструкции TRANSFORM указывается поле и групповая функция, применимая к нему. Данное поле выводится на пересечении строк и столбцов;

– в SELECT указывается поле, выводимое в заголовках строк;

– в FROM указываются имена таблиц, из которых выбираются данные;

– в конструкции GROUP BY указывается поле, по которому проводится группировка и которое выводится в качестве заголовков строк;

– в конструкции PIVOT указывается поле, значения которого выводятся в качестве заголовков столбцов.

* 1. **Перечислите и кратко охарактеризуйте операторы манипулирования данными.**

В операции манипулирования данными входят три действия: операция удаления записей – ей соответствует оператор **DELETE**, операция добавления или ввода новых записей – ей соответствует оператор **INSERT** и операция изменения (обновления полей в записях) – ей соответствует оператор **UPDATE**.

* 1. **Перечислите команды языка определения данных (Data Definition Language – DDL).**

CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE VIEW, DROP VIEW, ALTER VIEW, CREATE INDEX, DROP INDEX.

1. **ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БД**
   1. **Что такое хранимая запись?**

**Хранимая запись** – совокупность связанных элементов данных, соответствующих одной или нескольким логическим записям. Хранимая запись (физическая запись) состоит из двух частей: **служебной** части и **информационной**.

**Служебная часть** хранимой записи используется для идентификации записи, задания ее типа, хранения признака удаления, для хранения указателей на элементы записи, идентификатора длины записи, для установления структурных ассоциаций между записями, для кодирования значений элементов. Пользовательские программы не имеют доступа к служебной части физической записи.

Поля **информационной части** хранимой записи содержат значения элементов данных.

* 1. **Какие виды форматов хранимых записей Вы знаете?**

**а) позиционный способ организации хранимых записей**

В каждой записи порядок следования элементов одинаковый, тип элементов не указывается, хранится только значение. Поля имеют фиксированную длину, то есть значение элемента в каждом экземпляре записи появляется с одной и той же позиции, определенной в описании структуры файла базы данных. Все записи имеют одинаковую длину. При таком формате записи можно применять эффективные алгоритмы поиска (двоичный метод, метод золотого сечения и другие), но память используется не рационально. Если данные имеют меньшую длину, чем размер поля записи, то производится выравнивание влево или вправо, и свободные места заполняются пробелами. Количество свободных мест может быть сокращено путем использования разделителей или индексного метода.

**б) способ хранения записей с разделителем**

Запись имеет поля переменной длины и соответственно сама хранимая запись имеет переменную длину. Выбирается некоторый символ разделителя, который нигде больше при организации файлов базы данных не используется.

При таком способе организации хранимых записей память экономно расходуется, но невозможно применение быстрых алгоритмов поиска данных. Выигрываем в памяти, но проигрываем в быстродействии.

**в) индексный способ организации хранимых записей**

При данном способе для определения начала значений элементов записей используется массив указателей, размещенный в служебной части записи. Аналогично способу хранения с разделителем индексный способ более экономно расходует память, но не позволяет для поиска данных использовать эффективные алгоритмы поиска.

**г) способ хранения с описателями**

Этот способ эффективен, если запись содержит много элементов, значения которых отсутствуют и необходимо явно хранить только элементы с известными значениями.

* 1. **Перечислите методы доступа к данным.**

**Метод доступа к данным** – это совокупность технических и программных средств, обеспечивающих возможность хранения и выборки данных, расположенных на физических устройствах.

Выделяют три основных группы методов доступа к данным: **последовательные**, **индексные**, **произвольные** методы доступа.

* 1. **Что относится к группе последовательных методов? Дайте их краткую характеристику.**

В группе **последовательных** методов доступа к данным используется один файл базы данных. Перебор записей выполняется, начиная с первой, последовательно одна за другой. Записи не упорядочены и обычно имеют фиксированный размер. В зависимости от способа организации записей в файле различают последовательный метод доступа на смежной и несмежной памяти.

* 1. **Что относится к группе индексных методов? Дайте их краткую характеристику.**
  2. **Что относится к произвольным методам?**
  3. **В чем заключается метод хеширования идентификатора?**

**Метод хеширования идентификатора** - это процесс применения хеш-функции к уникальному идентификатору или ключу для получения фиксированной длины хеш-значения. Хеш-функция преобразует входные данные переменной длины в строку фиксированной длины, называемую хеш-кодом или хеш-значением.

* 1. **Что такое коллизия?**

**Коллизия** - случай преобразования ключа в уже занятый собственный адрес

1. **ТРАНЗАКЦИИ И ЦЕЛОСТНОСТЬ БАЗ ДАННЫХ**
   1. **Что такое транзакция?**

**Транзакция** - последовательность операторов манипулирования данными, выполняющаяся как единое целое и переводящая базу данных из одного целостного состояния в другое целостное состояние.

* 1. **Что понимают под целостностью базы данных?**

**Целостность базы данных** - свойство базы данных, означающее, что БД содержит полную и непротиворечивую информацию, необходимую для корректного функционирования приложений.

* 1. **Перечислите и охарактеризуйте свойства транзакций.**

Транзакция обладает четырьмя важными свойствами, известными как свойства ACID АСИД [10]:

− (А) Atomicity, **атомарность**. Транзакция выполняется как атомарная операция - либо выполняется вся транзакция целиком, либо она целиком не выполняется.

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

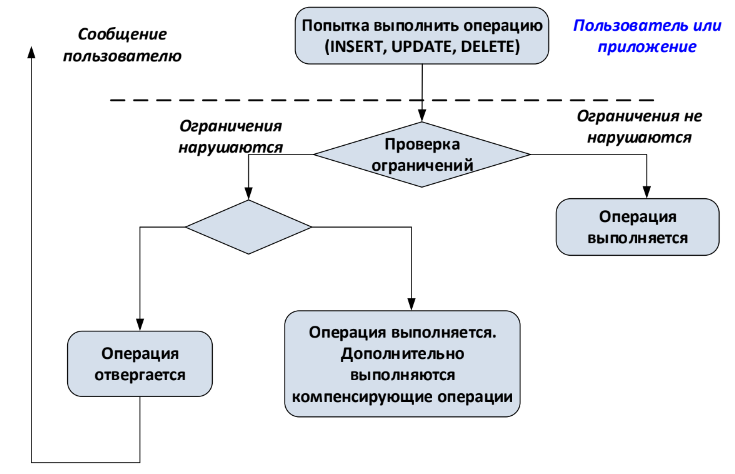
− (И) Isolation, **изоляция**. Конкурирующие за доступ к базе данных транзакции физически обрабатываются последовательно, изолированно друг от друга, но для пользователей это выглядит так, как будто они выполняются параллельно.

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

* 1. **Что такое ограничение целостности?**

**Ограничение целостности** – это некоторое утверждение, которое может быть истинным или ложным в зависимости от состояния базы данных.

* 1. **Приведите обобщенный алгоритм проверки ограничений.**

****

* 1. **Расскажите классификацию ограничений целостности по способам реализации.**

**− декларативная поддержка ограничений целостности** заключается в определении ограничений средствами языка определения данных (DDL – Data Definition Language).

**− процедурная поддержка ограничений целостности** заключается в использовании триггеров и хранимых процедур.

* 1. **Расскажите классификацию ограничений целостности по времени проверки.**

**− немедленно проверяемые ограничения** проверяются непосредственно в момент выполнения операции, могущей нарушить ограничение.

**− ограничения с отложенной проверкой**  – это ограничения на базу данных, а не на какие-либо отдельные операции. По умолчанию такие ограничения проверяются при конце транзакции.

* 1. **Расскажите классификацию ограничений целостности по области действия.**

**− ограничения домена** представляют собой ограничения, накладываемые только на допустимые значения домена.

Например, ограничением домена «Возраст сотрудника» может быть условие «Возраст сотрудника не менее 18 и не более 65».

Проверка ограничения. Ограничения домена сами по себе не проверяются. Если на каком-либо домене основан атрибут, то ограничение соответствующего домена становится ограничением этого атрибута.

**− ограничения атрибута** представляют собой ограничения, накладываемые на допустимые значения атрибута вследствие того, что атрибут основан на каком-либо домене. Ограничение атрибута в точности совпадают с ограничениями соответствующего домена. Отличие ограничений атрибута от ограничений домена в том, что ограничения атрибута проверяются.

Проверка ограничения. Ограничение атрибута является немедленно проверяемым ограничением. Действительно, ограничение атрибута не зависит ни от каких других объектов базы данных, кроме домена, на котором основан атрибут. Поэтому никакие изменения в других объектах не могут повлиять на истинность ограничения.

**− ограничения кортежа** представляют собой ограничения, накладываемые на допустимые значения отдельного кортежа отношения, и не являющиеся ограничением целостности атрибута. Требование, что ограничение относится к отдельному кортежу отношения, означает, что для его проверки не требуется никакой информации о других кортежах отношения.

Например, атрибут «Возраст сотрудника» в таблице «Спецподразделение», может иметь дополнительное ограничение «Возраст сотрудника не менее 25 и не более 45», помимо того, что этот атрибут уже имеет ограничение, определяемое доменом – «Возраст сотрудника не менее 18 и не более 65».

Проверка ограничения. Ограничение кортежа является немедленно проверяемым ограничением. Действительно, ограничение кортежа не зависит ни от каких других объектов базы данных, кроме атрибутов, входящих в состав кортежа. Поэтому никакие изменения в других объектах не могут повлиять на истинность ограничения.

**− ограничения отношения** представляют ограничения, накладываемые только на допустимые значения отдельного отношения, и не являющиеся ограничением целостности кортежа. Требование, что ограничение относится к отдельному отношению, означает, что для его проверки не требуется информации о других отношениях (в том числе не требуется ссылок по внешнему ключу на кортежи этого же отношения).

Например, ограничение целостности, определяемое требованием, что некоторая таблица должна быть не пуста, являются ограничениями отношения.

Проверка ограничения. К моменту проверки ограничения отношения должны быть проверены ограничения целостности кортежей этого отношения. Ограничение отношения может быть как немедленно проверяемым ограничением, так и ограничением с отложенной проверкой.

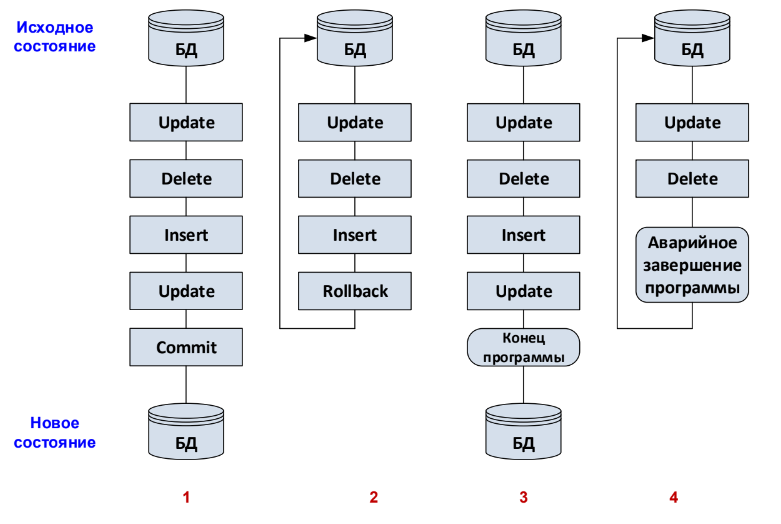
**− ограничения базы данных** представляют ограничения, накладываемые на значения двух или более связанных между собой отношений (в том числе отношение может быть связано само с собой).

Например, ограничение целостности ссылок, задаваемое внешним ключом отношения, является ограничением базы данных.

Проверка ограничения. К моменту проверки ограничения базы данных должны быть проверены ограничения целостности отношений. Ограничение базы данных может быть как немедленно проверяемым ограничением, так и ограничением с отложенной проверкой.

1. **ЖУРНАЛИЗАЦИЯ**
   1. **В чем заключается модель транзакций ANSI/ISO?**

Транзакция начинается с первого SQL-оператора, инициируемого пользователем или содержащегося в программе, изменяющего текущее состояние БД. Все по следующие SQL-операторы составляют тело транзакции. Транзакция завершается одним из четырех возможных путей

****

1) оператор COMMIT означает успешное завершение транзакции; его использование делает постоянными изменения, внесенные в базу данных в рамках текущей транзакции;

2) оператор ROLLBACK прерывает транзакцию, отменяя изменения, сделанные в базе данных в рамках этой транзакции; новая транзакция начинается непосредственно после использования ROLLBACK;

3) успешное завершение программы, в которой была инициирована текущая транзакция, означает успешное завершение транзакции (как будто был использован оператор COMMIT);

4) ошибочное завершение программы прерывает транзакцию (как будто был использован оператор ROLLBACK).

В этой модели каждый оператор, который изменяет состояние БД, рассматривается как транзакция, поэтому при успешном завершении этого оператора БД переходит в новое устойчивое состояние.

* 1. **Что такое журнал транзакций?**

**Журнал транзакций** - записи всех событий, происходящих с базой данных

* 1. **Перечислите принципы восстановления после сбоя.**

− результаты зафиксированных транзакций должны быть сохранены в восстановленном состоянии базы данных;

− результаты незафиксированных транзакций должны отсутствовать в восстановленном состоянии базы данных.

* 1. **Что такое индивидуальный откат транзакции?**

- стандартной ситуацией отката транзакции является ее явное завершение оператором ROLLBACK;

- аварийное завершение работы прикладной программы;

- принудительный откат транзакции в случае взаимной блокировки при параллельном выполнении транзакций.

* 1. **Каков механизм восстановления при индивидуальном откате?**

− Выбирается очередная запись из списка данной транзакции.

− Выполняется противоположная по смыслу операция: вместо операции INSERT выполняется соответствующая операция DELETE, вместо операции DELETE выполняется INSERT и вместо прямой операции UPDATE обратная операция UPDATE, восстанавливающая предыдущее состояние объекта базы данных.

− Любая из этих обратных операций также заносится в журнал.

− При успешном завершении отката в журнал заносится запись о конце транзакции.

* 1. **Что такое мягкий сбой?**

- при аварийном выключении электрического питания;

- при возникновении неустранимого сбоя процессора (например, срабатывании контроля оперативной памяти) и т. д. Ситуация характеризуется потерей той части базы данных, которая к моменту сбоя содержалась в буферах оперативной памяти.

* 1. **Каков механизм восстановления при мягком сбое?**

Необходимо восстановить содержимое БД по содержимому журналов транзакций, хранящихся на дисках.

* 1. **Что такое жесткий сбой?**

Восстановление после поломки основного внешнего носителя базы данных

* 1. **Каков механизм восстановления при жестком сбое?**

− по журналу в прямом направлении выполняются все операции;

− для транзакций, которые не закончились к моменту сбоя, выполняется откат.

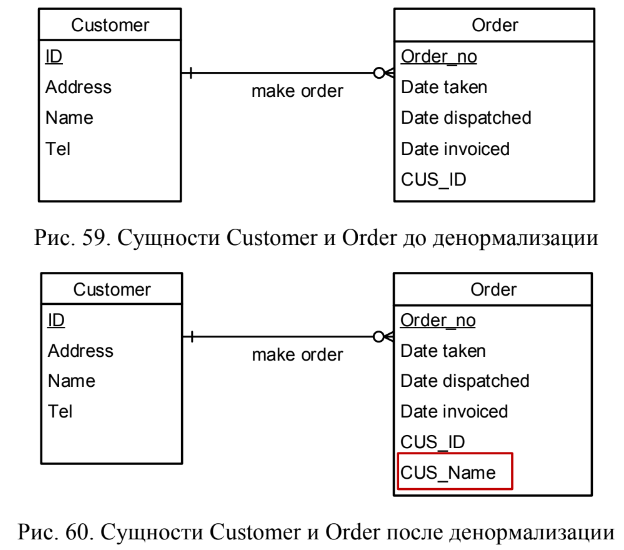
1. **ДЕНОРМАЛИЗАЦИЯ**
   1. **Что такое денормализация? Для чего применяют денормализацию?**

**Денормализация** - процесс достижения компромиссов в нормализованных таблицах посредством намеренного введения избыточности в целях увеличения производительности

**Денормализацию** применяют тогда, когда надо упростить процедуру чтения данных, а также — когда есть множество соединений таблиц (не надо будет лезть куда-то — все находится в одной таблице).

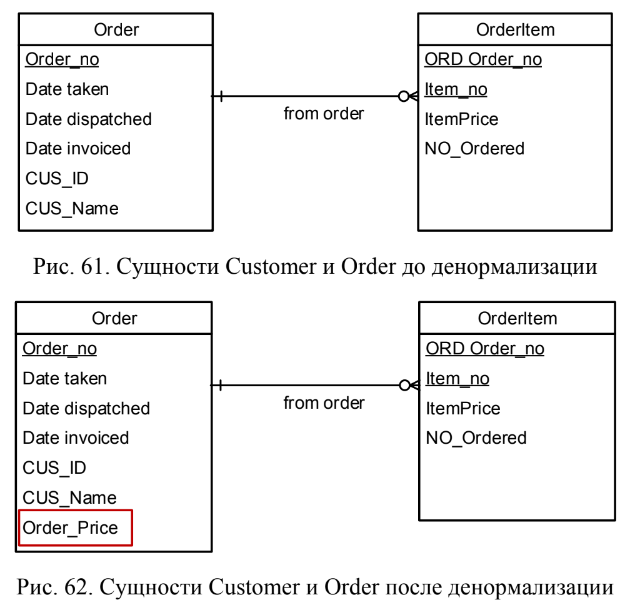
* 1. **В чем заключается нисходящая денормализация?**

**Нисходящая денормализация** предлагает перенос атрибута из одной (родительской) сущности в подчиненную (дочернюю) сущность (Рис.59 – Рис. 60). В денормализованной логической модели переместили фамилию клиента из сущности Customer (Клиент) в сущность Order (Заказ).



* 1. **В чем заключается восходящая денормализация?**

**Восходящая денормализация** предлагает перенос атрибута из подчиненной (дочерней) сущности в родительскую сущность, обычно в форме итоговых данных (Рис. 61 – Рис. 62).



* 1. **Что такое внутритабличная денормализация?**

**Внутритабличная денормализация** выполняется в пределах одной таблицы, т.е. это процесс введения избыточных колонок в одной таблице с целью увеличения производительности запроса строки по производному значению.

Например, если строка содержит две числовых колонки, X и Y, то значение Z, равное произведению X и Y (Z = X\*Y), легко вычислить во время выполнения.

* 1. **В чем заключается денормализация методом «разделяй и властвуй»?**

**Денормализация методом «разделяй и властвуй»** – это процесс разбиения нормализованной таблицы на две и более таблиц и создание между ними отношения «один к одному» с целью устранения дополнительных операций ввода/вывода или по техническим причинам.

* 1. **В чем заключается денормализация методом слияния таблиц?**

**Денормализация методом слияния таблиц** – это процесс объединения одной или более нормализованных таблиц с целью устранения операций соединений или уменьшения в некоторых случаях числа операций вставки.

* 1. **Перечислите наиболее распространенные методы реализации денормализации.**

**Разбиение** (splitting) таблиц является одним из общих методов денормализации, который применяется в физическом проектировании реляционных баз данных. Разбиение таблиц бывает двух видов – **вертикальное** разбиение и **горизонтальное** разбиение.

* 1. **Чем вертикальное разбиение отличается от горизонтального?**

**Вертикальное** разбиение является процессом перемещения некоторых колонок таблицы в другую новую таблицу, которая имеет тот же первичный ключ, что и исходная таблица.

**Горизонтальное** разбиение является процессом перемещения некоторых строк одной таблицы в другую новую таблицу, которая имеет такую же внутреннюю структуру, что и исходная таблица.

1. **РАСПРЕДЕЛЕННЫЕ БД**
   1. **Что такое распределенная БД?**

**Распределенная БД** - это набор отношений, хранящихся в разных узлах компьютерной сети и логически связанных таким образом, чтобы составлять единую совокупность данных

* 1. **Перечислите принципы распределенной БД, сформулированные Дейтом.**

1) **Локальная автономия**. Это качество означает, что управление данными на каждом из узлов распределенной системы выполняется локально.

2) **Независимость узлов**. Все узлы равноправны и независимы, а расположенные на них БД являются равноправными поставщиками данных в общее пространство данных.

3) **Непрерывность операций**. Это возможность непрерывного доступа к данным в рамках распределенной БД вне зависимости от их расположения и вне зависимости от операций, выполняемых на локальных узлах.

4) **Прозрачность расположения**. Пользователь, обращающийся к БД, ничего не должен знать о реальном, физическом размещении данных в узлах информационной системы.

5) **Прозрачная фрагментация**. Возможность распределенного (т.е. на различных узлах) размещения данных, логически представляющих собой единое целое. Существует фрагментация двух типов: горизонтальная и вертикальная. Первая означает, что строки таблицы хранятся на различных узлах. Вторая означает распределение столбцов логической таблицы по нескольким узлам.

6) **Прозрачное тиражирование**. Тиражирование данных – это асинхронный процесс переноса изменений объектов исходной базы данных в базы, расположенные на других узлах распределенной системы.

7) **Обработка распределенных запросов**. Возможность выполнения операций выборки данных из распределенной БД, посредством запросов, сформулированных на языке SQL.

8) **Обработка распределенных транзакций**. Возможность выполнения операций обновления распределенной базы данных, не нарушающих целостность и согласованность данных.

9) **Независимость от оборудования**. Это свойство означает, что в качестве узлов распределенной системы могут выступать компьютеры любых моделей и производителей.

10) **Независимость от операционных систем**. Это качество вытекает из предыдущего и означает многообразие операционных систем, управляющих узлами распределенной системы.

11) **Прозрачность сети**. Доступ к любым базам данных осуществляется по сети. Спектр поддерживаемых конкретной СУБД сетевых протоколов не должен быть ограничением системы, основанной на распределенной БД.

12) **Независимость от СУБД**. Это качество означает, что в распределенной системе могут работать СУБД различных производителей, и возможны операции поиска и обновления в базах данных различных моделей и форматов.

* 1. **Какие способы распределения БД Вы знаете?**

Существуют **централизованный**, **децентрализованный** и **смешанный** способы распределения данных.

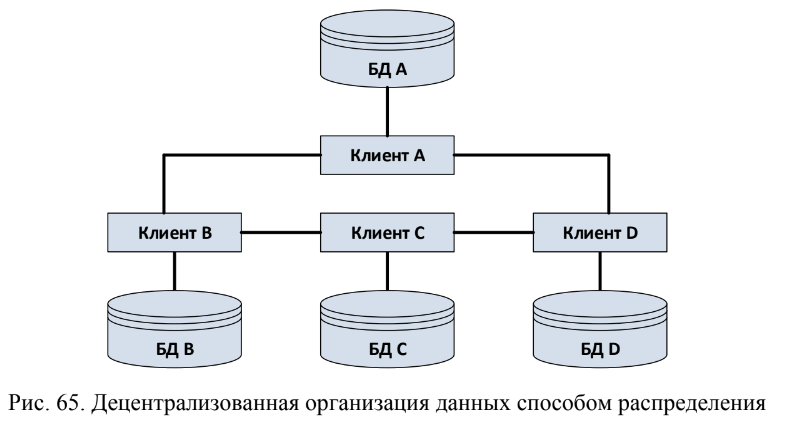
* 1. **В чем заключается идея централизованной организации данных? В чем ее сильные и слабые стороны?**

**Централизованная организация данных** является самой простой для реализации. На одном сервере находится единственная копия базы данных. Все операции с базой данных обеспечиваются этим сервером. Доступ к данным выполняется с помощью удалённого запроса или удалённой транзакции.

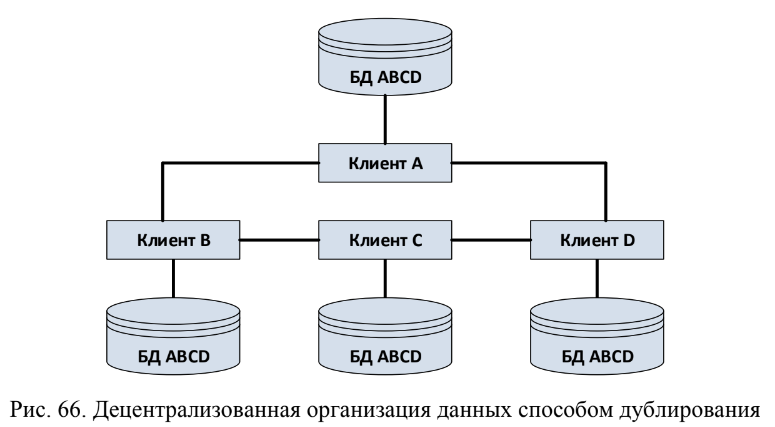
**Достоинством** такого способа является лёгкая поддержка базы данных в актуальном состоянии, а **недостатком** – то, что размер базы ограничен размером внешней памяти: все запросы направляются к единственному серверу с соответствующими затратами на стоимость связи и временную задержку. Отсюда ограничение на параллельную обработку. База может быть недоступной для удалённых пользователей при появлении ошибок связи и полностью выходит из строя при отказе центрального сервера.

* 1. **В чем заключается идея децентрализованной организации данных? В чем ее сильные и слабые стороны?**

**Децентрализованная организация данных** предполагает разбиение информационной базы на несколько физически распределенных. Каждый клиент пользуется своей базой данных, которая может быть либо частью общей информационной базы (Рис. 65), либо копией информационной базы в целом (Рис. 66), что приводит к ее дублированию для каждого клиента.

**Распределение данных на основе разбиения**. База данных размещается на нескольких серверах. Существование копий отдельных частей недопустимо. **Достоинства** этого метода: большинство запросов удовлетворяются локальными базами, что сокращает время ответа; увеличивается доступность данных и надёжность их хранения; стоимость запросов на выборку и обновление снижается по сравнению с централизованным распределением; система останется частично работоспособной, если выйдет из строя один сервер. **Недостатки**: часть удалённых запросов или транзакций может потребовать доступ ко всем серверам, что увеличивает время ожидания и цену обслуживания; необходимо иметь сведения о размещении данных в различных БД. Однако доступность и надёжность увеличиваются. Такие базы данных наиболее подходят к случаю совместного использования локальных и глобальных компьютерных сетей. 

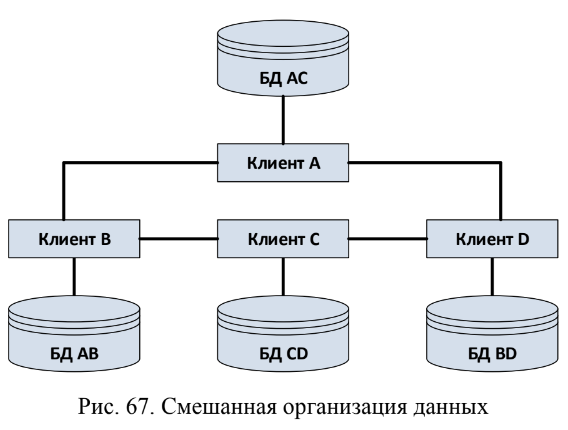
**Распределение данных на основе дублирования** (Рис. 66) заключается в том, что в каждом сервере сети размещается полная база данных. Это обеспечивает наибольшую надёжность хранения данных. **Недостатки**: повышенные требования к объему внешней памяти; усложнение корректировки баз, так как требуется синхронизация в целях согласования копий.

**Достоинство**: все запросы выполняются локально, что обеспечивает быстрый доступ. Данный способ используется, когда фактор надёжности является критическим, база небольшая, интенсивность обновления невелика. 

* 1. **В чем заключается идея смешанной организации хранения данных? В чем ее сильные и слабые стороны?**

**Смешанная организация хранения данных** объединяет два способа распределения: разбиение и дублирование (Рис. 67), приобретая при этом и преимущества и недостатки обоих способов.

Появляется необходимость хранить информацию о том, где находятся данные в сети. При этом достигается компромисс между объемом памяти под базу в целом и под базу в каждом сервере, чтобы обеспечить надёжность и эффективность её работы; легко реализуется параллельная обработка, то есть обслуживание распределённого запроса, или транзакции. Несмотря на гибкость смешанного способа организации данных, остаётся проблема взаимозависимости факторов, влияющих на производительность системы, проблема её надёжности и выполнения требований к памяти. Смешанный способ организации данных можно использовать лишь при наличии сетевой СУБД.



* 1. **В чем заключается идея метода построения распределенной БД «сверху вниз»?**

Проектирование РаБД «сверху-вниз» осуществляется в целом аналогично проектированию централизованных баз данных. В идеале оно проводится с помощью одной из формальных методологий, которые включают создание концептуальной модели базы данных, отображение ее в логическую модель данных и, наконец, создание (и настройку) специфических для конкретной СУБД структур.

* 1. **В чем заключается идея метода построения распределенной БД «снизу вверх»?**

Идея метода построения распределенной базы данных (РБД) "снизу вверх" заключается в том, чтобы начинать с мелких, локальных компонентов данных, а затем объединять их в более крупные структуры и распределенные системы. В этом методе акцент делается на построении системы отдельных, автономных узлов, которые затем объединяются для создания распределенной сети.

* 1. **Перечислите преимущества и недостатки распределенных СУБД.**

**Преимущества распределенных СУБД:**

1) отражение структуры организации;

2) разделяемость и локальная автономность;

3) повышение доступности данных;

4) повышение надежности;

5) повышение производительности;

6) экономические выгоды;

7) модульность системы;

**Недостатки распределенных СУБД:**

1) Повышение сложности;

2) Увеличение стоимости;

3) Проблемы защиты;

4) Усложнение контроля за целостностью данных;

5) Отсутствие стандартов;

6) Недостаток опыта;

7) Усложнение процедуры разработки;

8) Сложность управления и обусловленная этим потенциальная опасность потери целостности данных.