



Автономная некоммерческая профессиональная образовательная организация
«МЕЖДУНАРОДНЫЙ ВОСТОЧНО-ЕВРОПЕЙСКИЙ КОЛЛЕДЖ»

Пушкинская ул., д. 268, 426008, г. Ижевск. Тел.: (3412) 77-68-24. E-mail: mveu@mveu.ru, www.mveu.ru
ИНН 1831200089. ОГРН 1201800020641

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

по выполнению практических работ

при изучении профессионального модуля

ПМ.11 Разработка, администрирование и защита баз данных

по специальности

09.02.07 Информационные системы и программирование

Ижевск, 2023

Практическая работа – небольшой научный отчет, обобщающий проведенную учащимся работу, которую представляют для защиты преподавателю.

В процессе практического занятия учащиеся выполняют одну или несколько практических работ (заданий) под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

Состав и содержание практических занятий направлены на реализацию Государственных требований.

Наряду с формированием умений и навыков в процессе практических занятий обобщаются, систематизируются, углубляются и конкретизируются теоретические знания, вырабатывается способность и готовность использовать теоретические знания на практике, развиваются интеллектуальные умения.

Практические занятия проводятся в форме практической подготовки в виде работ, связанных с будущей профессиональной деятельностью.

К практическим работам предъявляется ряд требований, основным из которых является полное, исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения заданий и профессиональной подготовке учащихся.

I. Практические работы:

Тема практической работы № 1. «Сбор и анализ информации», объем часов 4

У1 работать с современными case-средствами проектирования баз данных;

У2 проектировать логическую и физическую схемы базы данных;

Цель практической работы приобретение практических навыков анализа предметной области, информационных задач и построения концептуальной модели базы данных.

Задание(я) Фирма выполняет ремонт компьютеров. Требуется разработать базу данных для хранения информации о выполнении ремонтных работ сотрудниками фирмы. При оформлении заказа фиксируется дата выполнения заказа, вид выполненной работы, исполнитель работы. Каждый исполнитель получает фиксированный процент вознаграждения от стоимости выполнения работы. Этот процент устанавливается персонально каждому исполнителю при заключении трудового договора между фирмой и работником.

Исполнитель получает вознаграждение, которое вычисляется как Стоимость выполнения заказа * Фиксированный процент вознаграждения.

Анализ описания предметной области позволяет выделить набор данных, которые должны храниться в проектируемой базе данных:

- Фамилия исполнителя работы;
- Имя исполнителя работы;
- Отчество исполнителя работы;
- Процент вознаграждения (может различаться для разных исполнителей);
- Наименование работы;
- Стоимость работы (фиксированная для каждого наименования работы);
- Дата исполнения работы

Исходя из набора данных, которые должны храниться в БД, можно выделить два информационных объекта: Исполнитель (Фамилия, Имя, Отчество, Процент вознаграждения) и Работа (Наименование, Стоимость работы). Определим соответствующие таблицы ИСПОЛНИТЕЛИ и РАБОТЫ (рис.1). Ни одно из первоначально заданных полей таблицы ИСПОЛНИТЕЛИ не определяет однозначно каждую запись таблицы, поэтому в таблицу введено поле Код исполнителя, значения в котором будут уникальными для каждого исполнителя. Это поле является первичным ключом таблицы ИСПОЛНИТЕЛИ и будет определено как ключевое поле. С этой же целью в таблицу РАБОТЫ введен первичный ключ Код работы. Структурированная таблица (рис.2)

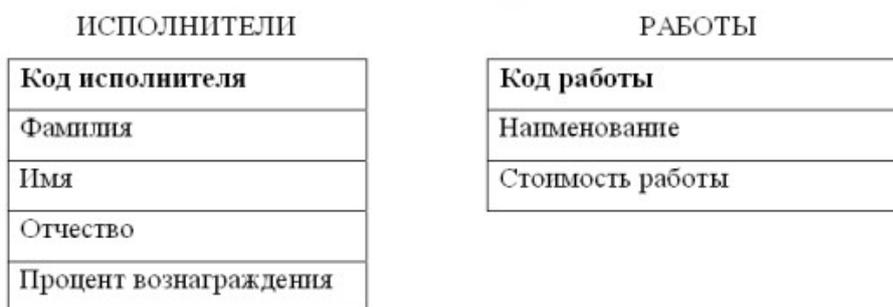


Рисунок 1

В таблице ИСПОЛНИТЕЛИ будут храниться записи вида:

| | | | | |
|---|----------|--------|-----------|----|
| 1 | Иванов | Андрей | Петрович | 20 |
| 2 | Алексеев | Игорь | Андреевич | 25 |

В таблице РАБОТЫ будут храниться записи вида:

| | | |
|---|---------------------------|-----------|
| 1 | Установка микропроцессора | 100.00 р. |
| 2 | Замена вентилятора | 50.00 р. |

Рисунок 2

Методические указания по ходу выполнения работы

При проектировании системы обработки данных именно данные и интересуют нас в первую очередь. Причём больше всего нас интересует организация данных. Помочь понять организацию данных, призвана информационная модель реального мира, на которой и основана система автоматизированной обработки данных.

Чтобы понять процесс построения информационной модели, необходимо знать ряд терминов, которые применяются при описании и представлении данных:

База данных - (БД, database) - поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Предметная область - некоторая часть реально существующей системы, функционирующая как самостоятельная единица. Полная предметная область может представлять собой экономику страны или группы союзных государств, однако на практике для информационных систем наибольшее значение имеет предметная область масштаба отдельного предприятия или корпорации.

Система управления базами данных (СУБД) - комплекс программных и языковых средств, необходимых для создания и модификации *базы данных*, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой.

Реляционная БД - основной тип современных *баз данных*. Состоит из таблиц, между которыми могут существовать связи по ключевым значениям.

Таблица *базы данных* (table) - регулярная структура, которая состоит из однотипных строк (записей, records), разбитых на столбцы (поля, fields).

В теории реляционных *баз данных* синоним таблицы - отношение (relation), в котором строка называется кортежем, а столбец называется атрибутом.

В концептуальной модели реляционной БД аналогом таблицы является сущность (entity), с определенным набором свойств - атрибутов, способных принимать определенные значения (набор допустимых значений - домен).

Ключевой элемент таблицы (ключ, regular key) - такое ее поле (простой ключ) или строковое выражение, образованное из значений нескольких полей (составной ключ), по которому можно определить значения других полей для одной или нескольких записей таблицы. На практике для использования ключей создаются индексы - служебная информация, содержащая упорядоченные сведения о ключевых значениях. В реляционной теории и концептуальной модели понятие "ключ" применяется для атрибутов отношения или сущности.

Первичный ключ (primary key) - главный ключевой элемент, однозначно идентифицирующий строку в таблице. Могут также существовать альтернативный (candidate key) и уникальный (unique key) ключи, служащие также для идентификации строк в таблице.

В реляционной теории первичный ключ - минимальный набор атрибутов, однозначно идентифицирующий кортеж в отношении.

В концептуальной модели первичный ключ - минимальный набор атрибутов сущности, однозначно идентифицирующий экземпляр сущности.

Связь (relation) - функциональная зависимость между объектами. В реляционных *базах данных* между таблицами устанавливаются связи по ключам, один из которых в главной (parent, родительской) таблице - первичный, второй - внешний ключ - во внешней (child, дочерней) таблице, как правило, первичным не является и образует связь "один ко многим" (1:M). В случае первичного внешнего ключа связь между таблицами имеет тип "один к одному" (1:1). Информация о связях сохраняется в базе данных.

Внешний ключ (foreign key) - ключевой элемент подчиненной (внешней, дочерней) таблицы, значение которого совпадает со значением первичного ключа главной (родительской) таблицы.

Ссылочная целостность данных (referential integrity) - набор правил, обеспечивающих соответствие ключевых значений в связанных таблицах.

Хранимые процедуры (stored procedures) - программные модули, сохраняемые в базе данных для выполнения определенных операций с информацией базы.

Триггеры (triggers) - хранимые процедуры, обеспечивающие соблюдение условий ссылочной целостности данных в операциях изменения первичных ключей (возможно каскадное изменение данных), удалении записей в главной таблице (каскадное удаление в дочерних таблицах) и добавлении записей или изменении данных в дочерних таблицах.

Репликация базы данных - создание копий базы данных (реплик), которые могут обмениваться обновляемыми данными или реплицированными формами, отчетами или другими объектами в результате выполнения процесса синхронизации.

Транзакция - изменение информации в базе в результате выполнения одной операции или их последовательности, которое должно быть выполнено полностью или не выполнено вообще. В СУБД существуют специальные механизмы обеспечения транзакций.

Язык SQL (Structured Query Language) - универсальный язык работы с базами данных, включающий возможности ее создания, модификации структуры, отбора данных по запросам, модификации информации в базе и прочие операции манипулирования базой данных.

Null - значение поля таблицы, показывающее, что информация в данном поле отсутствует. Разрешение на возможность существования значения Null может задаваться для отдельных полей таблицы.

Администратор базы данных (АБД) - лицо, или группа лиц, уполномоченных на ведение БД (модификация структуры и содержания БД, активизация доступа пользователей, выполнение других административных функций, которые затрагивают всех пользователей).

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

Внешняя схема - представление данных с точки зрения пользователя или прикладной программы.

Внутренняя схема - физическая структура данных.

Логическая структура БД - определение БД на физически независимом уровне.

Модель концептуальная (инфологическая) - описание предметной области, выполненное с использованием естественного языка, математических выражений, таблиц, графов и других средств, понятных всем людям, работающим над проектированием базы данных.

Модель физическая - определяющая размещение и способы поиска данных на внешних запоминающих устройствах СУБД.

Нормализация - представление сложных структур данных (документов) в виде двумерных таблиц (отношений).

Проектирование БД - упорядоченный формализованный процесс создания системы взаимосвязанных описаний - таких моделей предметной области, которые связывают (фиксируют) хранимые в базе данные с объектами предметной области, описываемые этими данными.

Распределенная БД - совокупность баз данных, которые обрабатываются и управляются по отдельности, а также могут разделять информацию.

Объектом называется элемент информационной системы. В реляционной теории баз данных объект называется также *сущностью*. Объект может быть реальным, например - человек, какой - либо предмет или населённый пункт, и абстрактным - событие, счёт покупателя или курс изучаемый студентами.

Классом объектов называется совокупность объектов, обладающих одинаковым набором свойств. Объекты и их свойства являются понятиями реального мира, В мире информации, существующем в представлении программиста свойства объектов называют атрибутами.

Атрибут - это информационное отображение свойств объекта. Например, клиент магазина, продающего автомобили, имеет такие атрибуты, как фамилию, имя, отчество, адрес, и возможно, идентификационный номер. Атрибут при реализации информационной модели на каком-либо носителе информации часто называют *элементом данных, полем данных* или просто *полем*.

Доменом называется набор записей данных одного типа, отвечающих поставленным условиям.

II. Классификация баз данных

По технологии обработки данных базы данных подразделяются на централизованные и распределенные.

Централизованная база данных хранится в памяти одной вычислительной системы. Эта вычислительная система может быть мэйнфреймом - тогда доступ к ней организуется с использованием терминалов - или файловым сервером локальной сети ПК.

Распределенная база данных состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей, которые хранятся в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По способу доступа к данным базы данных разделяются на **базы данных с локальным доступом** и **базы данных с сетевым доступом**.

Для всех современных баз данных можно организовать сетевой доступ с многопользовательским режимом работы.

Централизованные базы данных с сетевым доступом могут иметь такую архитектуру: следующую архитектуру:

- файл-сервер;
- клиент-сервер базы данных;
- "тонкий клиент" - сервер приложений - сервер базы данных



Рис.1. Схема работы с БД в локальной сети с выделенным файловым сервером (трехуровневая архитектура).

Файл-сервер. Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (файловый сервер). На этот компьютер устанавливается операционная система (ОС) для выделенного сервера (например, Microsoft Windows Server 2003). На нем же хранится совместно используемая централизованная БД в виде одного или группы файлов. Все другие компьютеры сети выполняют функции рабочих станций (могут работать в ОС Microsoft Windows 2000

Professional или Microsoft Windows 98). Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где и производится обработка информации. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также локальные БД на рабочих станциях.

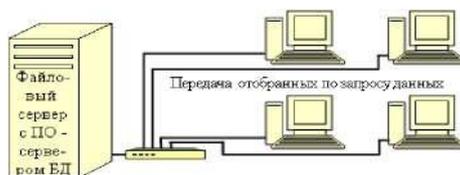


Рис. 2. Схема работы с БД в архитектуре "Клиент-сервер"

Клиент-сервер. В этой архитектуре на выделенном сервере, работающем под управлением серверной операционной системы, устанавливается специальное программное обеспечение (ПО) - сервер БД, например, Microsoft SQL Server или Oracle. СУБД подразделяется на две части: клиентскую и серверную. Основа работы сервера БД - использование языка запросов (SQL). Запрос на языке SQL, передаваемый клиентом (рабочей станцией) серверу БД, порождает поиск и извлечение данных на сервере. Извлеченные данные транспортируются по сети от сервера к клиенту. Тем самым, количество передаваемой по сети информации уменьшается во много раз.

Трехуровневая архитектура функционирует в Интранет- и Интернет-сетях. Клиентская часть ("тонкий клиент"), взаимодействующая с пользователем, представляет собой HTML- страницу в Web-браузере либо Windows-приложение, взаимодействующее с Web-сервисами. Вся программная логика вынесена на сервер приложений, который обеспечивает формирование запросов к базе данных, передаваемых на выполнение серверу баз данных. Сервер приложений может быть Web-сервером или специализированной программой (например, Oracle Forms Server).



Рис. 3. Схема работы с БД в трехуровневой архитектуре

По типу используемой модели данных. БД может быть основана на одной модели или на совокупности нескольких моделей. Любую модель

данных можно рассматривать как объект, который характеризуется своими свойствами (параметрами), и над ней, как над объектом, можно производить какие-либо действия.

Существуют три основных типа моделей данных - реляционная, иерархическая и сетевая.

Реляционная модель. Термин «реляционный» (от латинского relatio - отношение) указывает, прежде всего, на то, что такая модель хранения данных построена на взаимоотношении составляющих ее частей. В простейшем случае она представляет собой двухмерный массив или двухмерную таблицу, а при создании сложных информационных моделей составит совокупность взаимосвязанных таблиц. Каждая строка такой таблицы называется записью, а столбец - полем.

Реляционная модель данных имеет следующие свойства:

- Каждый элемент таблицы - один элемент данных.
- Все поля в таблице являются однородными, т.е. имеют один тип.
- Каждое поле имеет уникальное имя.
- Одинаковые записи в таблице отсутствуют.
- Порядок записей в таблице может быть произвольным и может характеризоваться количеством полей, типом данных.

| Номер сотрудника | Фамилия | Зарплата | Номер отдела |
|------------------|---------|----------|--------------|
| 1 | Иванов | 1000 | 1 |
| 2 | Петров | 2000 | 2 |
| 3 | Сидоров | 3000 | 1 |

Рис. 4 Реляционная модель данных

Иерархическая модель. Иерархическая модель БД представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое дерево (граф). Данная модель характеризуется такими параметрами, как уровни, узлы, связи. Принцип работы модели таков, что несколько узлов более низкого уровня соединяются при помощи связи с одним узлом более высокого уровня.

Узел - информационная модель элемента, находящегося на данном уровне иерархии.

Свойства иерархической модели данных:

- Несколько узлов низшего уровня связано только с одним узлом

высшего уровня.

- Иерархическое дерево имеет только одну вершину (корень), не подчиненную никакой другой вершине.
- Каждый узел имеет свое имя (идентификатор).
- Существует только один путь от корневой записи к более частной записи данных.

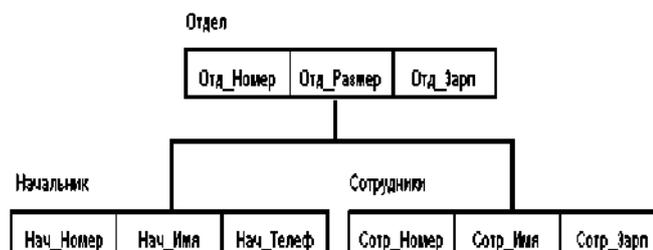


Рис.5 Иерархическая модель данных

Сетевая модель. Сетевая модель БД похожа на иерархическую. Она имеет те же основные составляющие (узел, уровень, связь), однако характер их отношений принципиально иной. В сетевой модели принята свободная связь между элементами разных уровней.

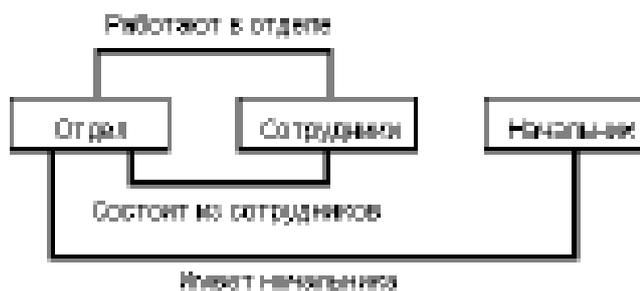


Рис. 6 Сетевая модель данных

III. Проектирование базы данных

Жизненный цикл БД. Как и любой программный продукт, база данных обладает собственным жизненным циклом (ЖЦБД). Главной составляющей в жизненном цикле БД является создание единой базы данных и программ, необходимых для работы. Жизненный цикл системы базы данных определяет и жизненный цикл всей информационной системы организации, поскольку база данных является фундаментальным компонентом информационной системы.

ЖЦБД включает в себя следующие основные этапы

- - планирование разработки базы данных;
- - определение требований к системе;

- - сбор и анализ требований пользователей;
- - проектирование базы данных:
концептуальное проектирование базы данных;
- логическое проектирование базы данных;
- физическое проектирование базы данных;
- разработка приложений:
- проектирование транзакций;
- проектирование пользовательского интерфейса;
- реализация;
- загрузка данных;
- тестирование;
- эксплуатация и сопровождение:
- анализ функционирования и поддержка исходного варианта БД;
- адаптация, модернизация и поддержка переработанных вариантов.

Здесь представлен перечень основных этапов ЖЦБД. Естественно, что конкретное наполнение каждого этапа в значительной степени зависит от сложности разрабатываемого продукта. Для малых же приложений с небольшим количеством пользователей и в некоторых других частных случаях, наоборот, жизненный цикл базы данных может быть значительно упрощен в результате корректировки содержания отдельных этапов.

Планирование разработки базы данных

Содержание данного этапа — разработка стратегического плана, в процессе которой осуществляется предварительное планирование конкретной системы управления базами данных. Общая информационная модель, созданная на этом шаге, должна быть вновь проанализирована и, если нужно, изменена на последующем этапе, этапе разработки проекта реализации.

Планирование разработки базы данных состоит в определении трех основных компонентов: объема работ, ресурсов и стоимости проекта. Планирование разработки базы данных должно быть связано с общей стратегией построения информационной системы организации.

Важной частью разработки стратегического плана является проверка осуществимости проекта, состоящая из нескольких частей.

Первая часть — проверка технологической осуществимости. Она состоит в выяснении вопроса, существует ли оборудование и программное обеспечение, удовлетворяющее информационным потребностям фирмы.

Вторая часть — проверка операционной осуществимости — выяснение наличия экспертов и персонала, необходимых для работы БД.



Рис 7. Этапы проектирования базы данных

Третья часть — проверка экономической целесообразности осуществления проекта. При исследовании этой проблемы весьма важно дать оценку ряду факторов, в том числе и таким:

- целесообразность совместного использования данных разными отделами
- величина риска, связанного с реализацией системы базы данных;
- ожидаемая выгода от внедрения подлежащих созданию приложений;
- время окупаемости внедренной БД;
- влияние системы управления БД на реализацию долгосрочных планов организации.

Планирование разработки баз данных также должно включать разработку стандартов, которые определяют, как будет осуществляться сбор данных, каким будет их формат, какая потребуется документация, как будет выполняться проектирование и реализация приложений.

Для поддержки планирования разработки базы данных может быть создана корпоративная модель данных, имеющая вид упрощенной ER-диаграммы.

Если результат проверки осуществимости проекта оказался положительным, можно перейти к определению требований к проекту.

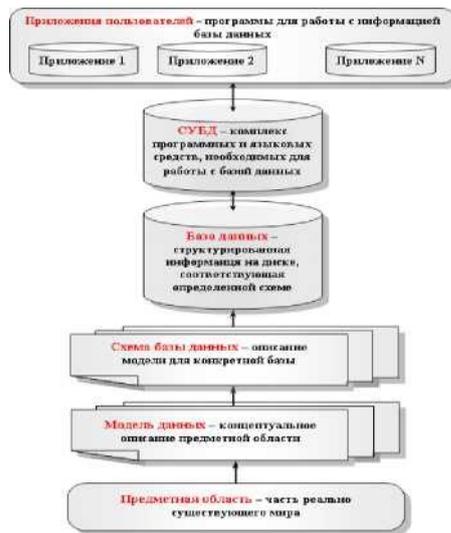


Рис.8. Области проектирования

Определение требований к системе

На данном этапе необходимо определить диапазон действия приложения базы данных, состав его пользователей и области применения.

Определение требований включает выбор целей БД, выяснение информационных потребностей различных отделов и руководителей фирмы и требований к оборудованию и программному обеспечению. При этом также требуется рассмотреть вопрос, следует ли создавать распределенную базу данных или же централизованную, и какие в рассматриваемой ситуации понадобятся коммуникационные средства. Написать краткий комментарий, описывающий цели системы.

Прежде чем приступать к проектированию приложения базы данных, важно установить границы исследуемой области и способы взаимодействия приложения с другими частями информационной системы организации. Эти границы должны охватывать не только текущих пользователей и области применения разрабатываемой системы, но и будущих пользователей и возможные области применения.

Сбор и анализ требований пользователей

Этот этап является предварительным этапом концептуального проектирования базы данных. Проектирование базы данных основано на информации о той части организации, которая будет обслуживаться базой данных.

Информационные потребности выясняются с помощью анкет, опросов менеджеров и работников фирмы, с помощью наблюдений за

деятельностью предприятия, а также отчетов и форм, которыми фирма пользуется в текущий момент.

На данном этапе необходимо создать для себя модель движения важных материальных объектов и уяснить процесс документооборота. По каждому документу необходимо установить периодичность использования, определить данные, необходимые для выполнения выделенных функций (анализируя существующую и планируемую документацию, выясняют, как получается каждый элемент данных, кем получается, где в дальнейшем используется, кем контролируется).

Самое пристальное внимание должно быть уделено дублированию информации, возможности появления ложной информации и причинам, которые ведут к их появлению. Также на этом этапе желательно представить общие параметры создаваемой базы.

В итоге собранная информация о каждой важной области применения приложения и пользовательской группе должна включать следующие компоненты: исходную и генерируемую документацию, подробные сведения о выполняемых транзакциях, а также список требований с указанием их приоритетов. На основании всей этой информации будут составлены спецификации требований пользователей в виде набора документов, описывающих деятельность предприятия с разных точек зрения.

Формализация собранной на этом этапе информации может быть повышена с помощью методов составления спецификаций требований, к числу которых относятся, например, технология структурного анализа и проектирования, диаграммы потоков данных и графики "вход — процесс — выход".

Поскольку системы с неадекватной или неполной функциональностью будут лишь раздражать пользователей, а чрезмерно увеличенный набор функциональных возможностей вызовет существенное усложнение системы, важность этого этапа в процессе разработки БД сложно переоценить.

Проектирование базы данных

Полный цикл разработки базы данных включает концептуальное, логическое и физическое ее проектирование.

Основными целями проектирования базы данных являются:

- представление данных и связей между ними, необходимых для всех основных областей применения данного приложения и любых существующих групп его пользователей;
 - создание модели данных, способной поддерживать выполнение любых требуемых транзакций обработки данных;
 - разработка предварительного варианта проекта, структура которого позволяет удовлетворить требования, предъявляемые к производительности системы.
- В создании БД как модели ПрО выделяют:
- объектную (предметную) систему, представляющую фрагмент реального мира;
 - информационную систему, описывающую некоторую объектную систему;
 - датологическую систему, представляющую информационную систему с помощью данных.

Оптимальная модель данных должна удовлетворять таким критериям, как: структурная достоверность, простота, выразительность, отсутствие избыточности, расширяемость, целостность, способность к совместному использованию.

Концептуальное проектирование базы данных

Первая фаза процесса проектирования базы данных заключается в создании для анализируемой части предприятия *концептуальной модели данных*. Построение ее осуществляется в определенном порядке: в начале создаются подробные модели пользовательских представлений данных; затем они интегрируются в концептуальную модель данных. Концептуальное проектирование приводит к созданию *концептуальной схемы* базы данных.

Существует два основных подхода к проектированию систем баз данных: "нисходящий" и "восходящий".

При восходящем подходе, который применяется для проектирования простых баз данных с относительно небольшим количеством атрибутов, работа начинается с самого нижнего уровня — уровня определения атрибутов, которые на основе анализа существующих между ними связей группируются в отношения. Полученные отношения в дальнейшем подвергаются процессу нормализации, который приводит к созданию нормализованных взаимосвязанных таблиц, основанных на функциональных зависимостях между атрибутами.

Проектирование сложных баз данных с большим количеством атрибутов, поскольку установить среди атрибутов все существующие

функциональные зависимости довольно затруднительно, осуществляется использованием нисходящего подхода. Начинается этот подход с разработки моделей данных, которые содержат несколько высокоуровневых сущностей и связей, затем работа продолжается в виде серии нисходящих уточнений низкоуровневых сущностей, связей и относящихся к ним атрибутов.

Нисходящий подход демонстрируется в концепции модели "сущность-связь" (EntityRelationship model — ER-модель) — самой популярной технологии высокоуровневого моделирования данных, предложенной П. Ченом.

Модель "сущность — связь" относится к семантическим моделям. *Семантическое моделирование* данных, связанное со смысловым содержанием данных, независимо от их представления в ЭВМ, изначально возникло с целью повышения эффективности и точности проектирования баз данных. Методы семантического моделирования оказались применимы ко многим пользовательским проблемам и легко преобразуемы в сетевые, иерархические и реляционные модели.

Помимо "нисходящего" и "восходящего" подходов, для проектирования баз данных могут применяться другие подходы, являющиеся некоторыми комбинациями указанных.

В построении *общей концептуальной модели данных* выделяют ряд этапов.

- Выделение локальных представлений, соответствующих обычно относительно независимым данным. Каждое такое представление проектируется как подзадача.

- Формулирование объектов, описывающих локальную предметную область проектируемой БД, и описание атрибутов, составляющих структуру каждого объекта.

- Выделение ключевых атрибутов.
- Спецификация связей между объектами. Удаление избыточных связей,

- Анализ и добавление не ключевых атрибутов.

- Объединение локальных представлений.

Построение концептуальной модели данных осуществляется на основе анализа описания предметной области на естественном языке, сделанного конечным пользователем. В процессе разработки концептуальная модель данных постоянно подвергается тестированию и проверке на соответствие требованиям пользователей. Созданная концептуальная модель

данных предприятия является источником информации для фазы логического проектирования базы данных.

Логическое проектирование базы данных

Цель второй фазы проектирования базы данных состоит в создании логической модели данных для исследуемой части предприятия.

Логическая модель, отражающая особенности представления о функционировании предприятия одновременно многих типов пользователей, называется *глобальной логической моделью данных*. Для создания глобальной логической модели данных предприятия можно выбрать один из двух основных подходов — централизованный подход или подход на основе интеграции представлений.

Отправным моментом при *централизованном подходе*, который применим только для не слишком сложных баз данных, является образование единого списка требований путем объединения требований всех типов пользователей.

При использовании *метода интеграции представлений* осуществляется слияние отдельных локальных логических моделей данных, отражающих представления разных групп пользователей, в единую глобальную логическую модель данных всего предприятия.

В дальнейшем процесс проектирования БД должен опираться на определенную модель данных (реляционная, сетевая, иерархическая), которая определяется типом предполагаемой для реализации информационной системы СУБД. После чего сама концептуальная модель данных уточняется и преобразуется в логическую модель данных.

В процессе разработки логическая модель данных должна постоянно подвергаться проверке как на соответствие требованиям пользователей, так и на отсутствие избыточности данных, способной вызвать в будущем аномалии обновления.

Построенная логическая модель данных в дальнейшем будет востребована на этапе физического проектирования, а также на этапе эксплуатации и сопровождения уже готовой системы, позволяя наглядно представить любые вносимые в базу данных изменения.

На этом шаге желательно создание следующих документов:

- набора подсхем;
- спецификаций для физического проектирования приложений;
- руководства по разработке программ (интерфейсы с пользователем и межпрограммные интерфейсы);
- руководства по сопровождению БД.

Концептуальное и логическое проектирование — это итеративные процессы, которые включают в себя ряд уточнений, продолжающиеся до тех пор, пока не будет получен наиболее соответствующий структуре предприятия продукт.

Физическое проектирование базы данных

Целью проектирования на данном этапе является создание описания СУБД ориентированной модели БД. Следует учитывать, что на этой стадии разработки возможны возвраты на более ранние этапы ЖЦБД. Например, решения, принимаемые на этапе физического проектирования с целью повышения производительности системы, могут привести к необходимости внести изменения в структуру логической модели данных.

Действия, выполняемые на этом этапе, слишком специфичны для различных моделей данных, поэтому их сложно обобщить. Остановимся на реляционной модели данных. В этом случае под физическим проектированием подразумевается:

- создание описания набора реляционных таблиц и ограничений для них на основе информации, представленной в глобальной логической модели данных;
- определение конкретных структур хранения данных и методов доступа к ним, обеспечивающих оптимальную производительность системы с базой данных;
- разработка средств защиты создаваемой системы.

Разработка приложений

Параллельно с проектированием системы базы данных выполняется разработка приложений. Главные составляющие данного процесса — это проектирование транзакций и пользовательского интерфейса.

Проектирование транзакций

Транзакции представляют некоторые события реального мира. Все транзакции должны обращаться к базе данных с той целью, чтобы хранимые в ней данные всегда гарантированно соответствовали текущей ситуации в реальном мире.

Транзакция может состоять из нескольких операций, однако с точки зрения пользователя эти операции представляют собой единое целое, переводящее базу данных из одного непротиворечивого состояния в другое. Реализация транзакций опирается на тот факт, что СУБД способна обеспечивать сохранность внесенных во время транзакции изменений в БД и непротиворечивость базы данных даже в случае возникновения сбоя. Для незавершенной транзакции СУБД гарантирует отмену всех внесенных ею изменений.

Проектирование транзакций заключается в определении:

- данных, которые используются транзакцией;
- функциональных характеристик транзакции;
- выходных данных, формируемых транзакцией;
- степени важности и интенсивности использования транзакции.

Существует три основных типа транзакций: извлечения, обновления и смешанные транзакции.

Транзакции извлечения используются для выборки некоторых данных с целью отображения их на экране или помещения в отчет.

Транзакции обновления используются для вставки новых, удаления старых или коррекции существующих записей базы данных.

Смешанные транзакции включают как операции извлечения, так и операции обновления данных.

Транзакции могут представлять собой сложные операции, которые раскладываются на несколько более простых операций, каждая из которых представляет собой отдельную транзакцию.

Проектирование пользовательского интерфейса

Пользовательский интерфейс приложений базы данных является одним из важнейших компонентов системы. Интерфейс должен быть удобным и обеспечивать все функциональные возможности, предусмотренные в спецификациях требований пользователей.

Специалисты рекомендуют при проектировании пользовательского интерфейса использовать следующие основные элементы и их характеристики:

- легко узнаваемые названия полей;
- согласованную терминологию и сокращения;

- удобные средства перемещения курсора;
- средства исправления отдельных ошибочных символов и целых полей;
- средства вывода сообщений об ошибках при вводе недопустимых значений;
- особое выделение необязательных для ввода полей;
- средства вывода пояснительных сообщений с описанием полей;
- средства вывода сообщения об окончании заполнения формы.

Реализация

На данном этапе осуществляется физическая реализация базы данных и разработанных приложений, позволяющих пользователю формулировать требуемые запросы к БД и манипулировать данными в БД.

Тема практической работы № 2 «Проектирование реляционной схемы базы данных в среде СУБД» объем часов 4

У1 работать с современными case-средствами проектирования баз данных;

У2 проектировать логическую и физическую схемы базы данных;

Цель практической работы приобретение практических навыков анализа предметной области, информационных задач и построения концептуальной модели базы данных.

Задание(я) по заданному описанию предметной области построить концептуальную модель базы данных:

- Выделите типы сущностей;
- Выделите типы связей и определите для них показатели кардинальности и степень участия сторон;
- Выделите атрибуты и свяжите их типами сущностей и связей;
- Определите потенциальные и первичные ключи сущностей;
- Нарисуйте ER-диаграмму и проанализируйте информационные задачи и группы пользователей.

Вариант 1.

Задача - организация учебного процесса в вузе:

* Студенты: паспортные данные, адрес, дата зачисления, номер приказа, факультет, группа, является ли старостой, кафедра (специализация), изучаемые (изученные) предметы, оценки, задолженности, стипендия.

* Учебные курсы: название, факультет(ы), групп(ы), кафедра, семестр(ы), форма отчётности, число часов.

* Преподаватели: паспортные данные, адрес, телефон, фотография, кафедра, должность, учёная степень, начальник (зав. кафедрой), предмет(ы), число ставок, зарплата.

Вариант 2.

Учет и выдача книг в библиотеке вуза:

* Книги: авторы, название, раздел УДК, раздел (техническая, общественно-политическая и т.п.), место и год издания, издательство, количество страниц, иллюстрированность, цена, дата покупки, номер сопроводительного документа (чек, счёт/накладная), вид издания (книги, учебники, брошюры, периодические издания), инвентарный номер (есть только для книг и некоторых учебников), длительность использования читателями (год, две недели, день), электронная версия книги или ее реферата (отсканированный текст).

* Читатели: номер читательского билета, ФИО, год рождения, адрес, дата записи, вид (студент, аспирант, преподаватель, сотрудник), курс, номер группы, названия взятых книг и даты их выдачи.

Вариант 3.

Отдел кадров некоторой компании.

* Сотрудники: ФИО, паспортные данные, фотография, дом. и моб. телефоны, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, тип(ы) работы, задание(я), проект(ы), размер зарплаты, форма зарплаты (почасовая, фиксированная).

* Отделы: название, комната, телефон(ы), начальник, размер финансирования, число сотрудников.

* Проекты: название, дата начала, дата окончания, размер финансирования, тип финансирования (периодический, разовый), задачи и их исполнители, структура затрат и статьи расходов.

Вариант 4.

Отдел поставок некоторого предприятия:

* Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.

Сырьё: тип, марка, минимальный запас на складе, время задержки, цена, продукты, при производстве которых используется, потребляемые объёмы (необходимый, реальный, на единицу продукции)

Методические указания по ходу выполнения работы

Проектирование базы данных (БД) - одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

- корректность схемы БД, т.е. база должна быть гомоморфным образом моделируемой предметной области (ПО), где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу - адекватные процедуры обработки данных;
- обеспечение ограничений ;
- эффективность функционирования ;
- защита данных (от аппаратных и программных сбоев и несанкционированного доступа);
- простота и удобство эксплуатации;
- гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

Внимание! Базы данных всегда проектируются под конкретное назначение системы.

Техника проектирования баз данных может измениться в целом и в деталях в зависимости от назначения системы. Например, следует различать проектирование систем складирования данных и проектирование так называемых OLTP-систем, ориентированных на оперативную обработку транзакций. В данном учебном курсе рассматривается проектирование баз данных в основном для OLTP-систем. Именно на таких системах исторически сложилась техника проектирования баз данных.

Этапы проектирования базы данных

Процесс проектирования включает в себя следующие этапы:

Концептуальное проектирование - это процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации.

Логическое проектирование - это процесс конструирования информационной модели на основе существующих моделей данных, не зависимо от используемой СУБД и других условий физической реализации.

Физическое проектирование - это процедура создания описания конкретной реализации БД с описанием структуры хранения данных, методов доступа к данным.

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО.

Концептуальная модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПО и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

Рассмотрим основные подходы к созданию концептуальной модели предметной области.

1. Функциональный подход к проектированию БД

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

2. Предметный подход к проектированию БД

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

3. Проектирование с использованием метода "сущность-связь"

Метод "сущность-связь" (entity-relation, ER-method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап инфологического проектирования начинается с моделирования ПО.

Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов ПО. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей.

Сущность - это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр - конкретными значениями свойств.

Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки - сильная сущность, а абонемент этого читателя - слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные - базовыми (основными, родительскими).

Для каждой сущности выбираются свойства (атрибуты). Различают:

- Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и включают в себе интересные свойства сущности.

- Составные и простые атрибуты. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным

типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

- Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).

- Основные и производные атрибуты. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности - множества значений (или домена), которые может принимать данный атрибут.

Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность- сущность", "сущность- атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность".

Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают факультативные и обязательные связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует обязательная связь (обозначается двойной линией). Иначе связь является факультативной.

По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:N) и "многие ко многим" (M:N).

Степень связи определяется количеством сущностей, которые охвачены данной связью. Пример бинарной связи - связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа экзамен между сущностями ДИСЦИПЛИНА, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это Дата проведения и Оценка). Пример ER-диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 3.

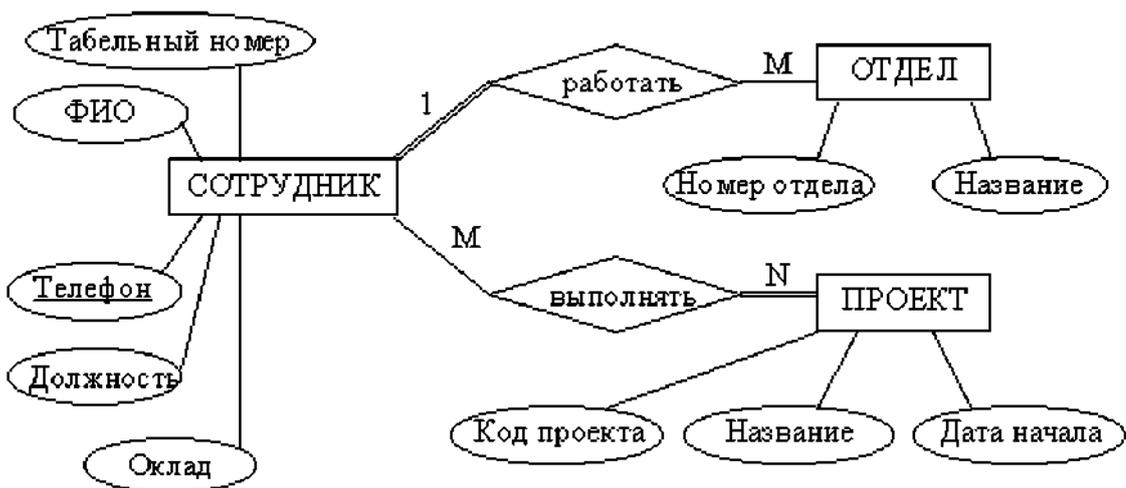


Рисунок 3

Анализ предметной области

В качестве примера возьмем базу данных компании, которая занимается издательской деятельностью.

База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников компании. БД должна содержать данные о сотрудниках компании, книгах, авторах, финансовом состоянии компании и предоставлять возможность получать разнообразные отчёты.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- каждая книга издаётся в рамках контракта;
- книга может быть написана несколькими авторами;
- контракт подписывается одним менеджером и всеми авторами книги;
- каждый автор может написать несколько книг (по разным контрактам);
- порядок, в котором авторы указаны на обложке, влияет на размер гонорара;
- если сотрудник является редактором, то он может работать одновременно над несколькими книгами;
- у каждой книги может быть несколько редакторов, один из них - ответственный редактор;
- каждый заказ оформляется на одного заказчика;
- в заказе на покупку может быть перечислено несколько книг.

Выделим базовые сущности этой предметной области:

1. Сотрудники компании. Атрибуты сотрудников - ФИО, табельный номер, пол, дата рождения, паспортные данные, ИНН, должность, оклад, домашний адрес и телефоны. Для редакторов необходимо хранить сведения о редактируемых книгах; для менеджеров - сведения о подписанных контрактах.

2. Авторы. Атрибуты авторов - ФИО, ИНН (индивидуальный номер налогоплательщика), паспортные данные, домашний адрес, телефоны. Для авторов необходимо хранить сведения о написанных книгах.

3. Книги. Атрибуты книги - авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар.

4. Контракты будем рассматривать как связь между авторами, книгами и менеджерами. Атрибуты контракта - номер, дата подписания и участники.

5. Для отражения финансового положения компании в системе нужно учитывать заказы на книги. Для заказа необходимо хранить номер заказа, заказчика, адрес заказчика, дату поступления заказа, дату его выполнения, список заказанных книг с указанием количества экземпляров.

ER-диаграмма издательской компании приведена на рис. 4 (базовые сущности на рисунке выделены полужирным шрифтом).

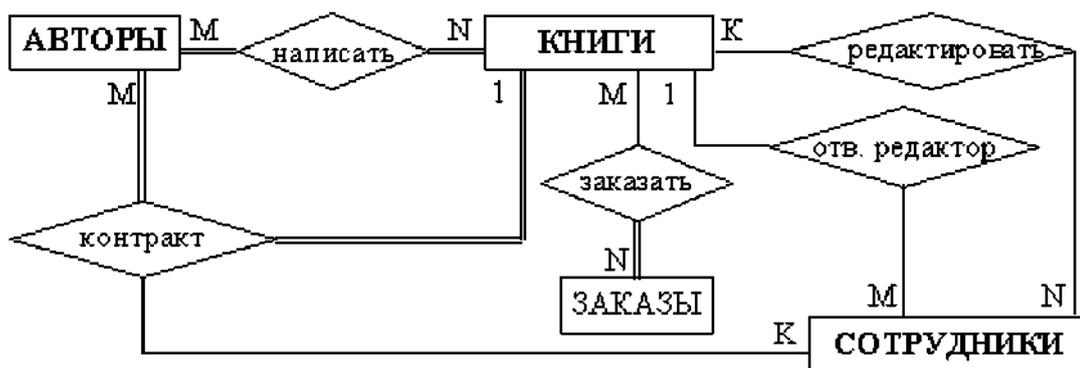


Рисунок 4

Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

- администрация (дирекция);
- менеджеры;
- редакторы;
- сотрудники компании, обслуживающие заказы.

Определим границы информационной поддержки пользователей:

- 1) Функциональные возможности:
 - ведение БД (запись, чтение, модификация, удаление в архив);
 - обеспечение логической непротиворечивости БД;
 - обеспечение защиты данных от несанкционированного или

случайного доступа (определение прав доступа);

- реализация наиболее часто встречающихся запросов в готовом виде;
- предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

2) Готовые запросы:

- получение списка всех текущих проектов (книг, находящихся в печати и в продаже);
- получение списка редакторов, работающих над книгами;
- получение полной информации о книге (проекте);
- получение сведений о конкретном авторе (с перечнем всех книг);
- получение информации о продажах (по одному или по всем проектам);
- определение общей прибыли от продаж по текущим проектам;
- определение размера гонорара автора по конкретному проекту.

II. Общие рекомендации

По всем вопросам, связанным с изучением дисциплины (включая самостоятельную работу), консультироваться с преподавателем.

III. Контроль и оценка результатов

Оценка за выполнение практической работы выставляется в форме по пятибалльной системе и учитывается как показатель текущей успеваемости студента.

| Качественная оценка индивидуальных образовательных достижений | | Критерии оценки результата |
|---|-------------------|---|
| балл (оценка) | вербальный аналог | |
| 5 | отлично | Представленные работы высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, выполнены все предусмотренные практической работой задания. |
| 4 | хорошо | Уровень выполнения работы отвечает всем требованиям, теоретическое содержание курса освоено полностью без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные практической работой задания выполнены, некоторые из выполненных заданий, возможно, содержат ошибки. |

| | | |
|---|----------------------|---|
| 3 | удовлетворительно | Уровень выполнения работы отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных практической работой заданий выполнено, некоторые виды заданий выполнены с ошибками. |
| 2 | не удовлетворительно | Теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных практической работой заданий не выполнено. |