



**Некоммерческое  
акционерное  
общество**

АЛМАТИНСКИЙ  
УНИВЕРСИТЕТ  
ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра казахского и  
русского языков

## **РУССКИЙ ЯЗЫК**

Методические указания и варианты по выполнению семестровых работ  
для студентов специальности 5В070200 – Автоматизация и управление

Алматы 2016

СОСТАВИТЕЛЬ: Р.А.Досмаханова. Русский язык. Методические указания и варианты по выполнению семестровых работ для студентов специальности 5В070200 – Автоматизация и управление. – Алматы: АУЭС, 2016. – 42 с.

В данную методическую разработку включены задания к шести семестровым работам, варианты текстов по специальности, перечень учебно-научной литературы и список периодических изданий для их выполнения.

Методические указания предназначены для студентов бакалавриата дневной формы обучения специальности 5В070200 – Автоматизация и управление

Библиогр. – 21 назв.

Рецензент: старший преподаватель С.Б. Бухина

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2016 г.

©НАО «Алматинский университет энергетики и связи», 2016 г.

## Введение

Предлагаемые методические указания составлены в соответствии с Типовой учебной программой дисциплины «Русский язык» (объем - 6 кредитов) и представляют варианты письменных заданий шести семестровых работ студентов.

Основная цель выполнения СРС заключается в выявлении уровня овладения обучающимися различными видами лингвистического анализа и продуцирования учебно-научных текстов, предусмотренных рабочей программой дисциплины.

В методических указаниях сформулированы темы, цели и задачи семестровых работ; представлены варианты текстов для анализа, списки рекомендуемой учебно-научной литературы.

Требования, предъявляемые к выполнению СРС:

1) Семестровая работа должна быть выполнена в соответствии с графиком выдачи и приёма СРС. Вариант текста для СРС № 2 и СРС № 3 определяется преподавателем.

2) Семестровая работа должна быть выполнена компьютерной вёрсткой шрифтом Times New Roman, кегль 14, одинарным междустрочным интервалом, в текстовом редакторе «MS Word». Абзацы в тексте начинают отступом для первой строки – 1,25 см. Размеры полей: верхнее - 2 см, нижнее – 2,5 см, левое – 2,5 см, правое - 1,8 см. Выравнивание текста – по ширине (подробную информацию об оформлении см. Lib.aipet.kz).

3) Титульный лист семестровой работы студента оформляется по образцу (см. СТ НАО 56023-1910-04-2014, Приложение С).

4) В конце работы должен быть приведён список использованной литературы и интернет-ресурсов. Последние ссылки должны быть конкретными с указанием даты обращения. Ссылки типа [www.yandex.ru](http://www.yandex.ru) или [www.google.ru](http://www.google.ru) не являются корректными.

5) В случае обнаружения плагиата к студенту могут быть применены санкции по усмотрению преподавателя (вплоть до аннулирования положительных результатов и получения оценки «неудовлетворительно» без права повторной сдачи).

## **Семестровая работа студента № 1**

**Тема:** функционально-смысловые типы речи.

**Цель:** проявить навыки различения функционально-смысловых типов речи (описание, повествование, рассуждение).

**Задачи:**

- 1) Подобрать три типа текста, относящиеся к трем функционально-смысловым типам речи.
- 2) Указать 2-3 признака конкретного типа речи в каждом тексте.
- 3) Составить толковый словарь незнакомых лексических единиц по каждому тексту (из каждого текста не менее 10-ти слов).

## **Рекомендуемая литература для выполнения СРС № 1**

1 Русский язык: учебное пособие для студентов казахских отделений университета (бакалавриат)/ Под ред. К.К.Ахмедьярова, Ш.К.Жаркынбековой. – Алматы: Қазақ университеті, 2009. – 226 с.

2 Мухамадиев Х.С. Пособие по научному стилю речи: для казахских отделений университета. - 3-е изд. – Алматы: Қазақ университеті, 2011.- 210 с.

3 Ишлинский А.Ю. Политехнический словарь. – М.: Советская энциклопедия, 1980. – 656 с.

4 Даль В.И. Большой иллюстрированный толковый словарь русского языка: современное написание. Около 1500 илл. – М.: «АСТ – Астрель - Хранитель», 2008. – 352 с.

## **Семестровая работа студента № 2**

**Тема:** структурно-смысловой анализ текста по специальности.

**Цель:** показать умение анализировать структурно-смысловое строение научного текста

**Задачи:**

- 1) Определить тему текста, выразив ее словом-темой и обозначив буквой «Т».
- 2) Определить коммуникативную задачу текста, в которой заключена данная информация текста, обозначив аббревиатурой «КЗТ».
- 3) Сделать анализ реализации КЗТ путем деления на микротемы (МТ-1, МТ-2, МТ-3 ...).
- 4) Выделить в тексте одно ССЦ и определить способы связи предложений в нем (последовательная и параллельная).

## **Рекомендуемая литература для выполнения СРС № 2**

1 Русский язык: учебное пособие для студентов казахских отделений университета (бакалавриат)/ Под ред. К.К.Ахмедьярова, Ш.К.Жаркынбековой. – Алматы: Қазақ университеті, 2009. – 226 с.

2 Мухамадиев Х.С. Пособие по научному стилю речи: для казахских отделений университета. - 3-е изд. – Алматы: Қазақ университеті, 2011.- 210 с.

3 Мейрбекова М.М., Хайрушева Е.Е. Пособие по русскому языку для студентов механико-математического факультета университета. – Алматы: Қазақ университеті, 2013. – 178 с.

## **Варианты текстов для выполнения СРС № 2 и СРС № 3**

### **Вариант 1**

### **Проектирование базы данных**

Развитие вычислительной техники оказало значительное влияние на многие стороны жизни общества. За последние десятилетия значительно расширились сферы применения вычислительной техники и соответствующий рост ее влияния на повседневную жизнь. Автоматизированная система, организующая данные и выдающая информацию, представляет собой информационную систему. Одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы, - проектирование базы данных. В результате её решения должны быть определены: содержание базы данных, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Процесс проектирования базы данных представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. В общем случае можно выделить следующие этапы проектирования: системный анализ и словесное описание информационных объектов предметной области; проектирование концептуальной модели предметной области; выбор конкретной системы управления базами данных; описание базы данных в терминах выбранной системы и, наконец, физическое проектирование базы данных, то есть выбор эффективного размещения базы данных на внешних носителях для обеспечения наиболее эффективной работы приложения.

Современная волна информационных технологий управления данными основывается на использовании систем управления реляционными базами данных. Все языки манипулирования данными, созданные до появления реляционных баз данных были ориентированы на операции с данными, представленными в виде логических записей файлов. Это требовало от

пользователей детального знания организации хранения данных и достаточных усилий для указания не только того, какие данные нужны, но и того, где они размещены, и как шаг за шагом получить их. Современные языки баз данных ориентированы в большей степени на конечный результат обработки данных, чем на процедуру этой обработки.

## Вариант 2

### Метрология

Метрология – научная основа государственной системы измерений. Метрология – это наука об измерениях, методах и средствах обеспечения их единства и требуемой точности. «Метро» - мера (греч.), «логос» - учение (греч.).

Современная метрология включает в себя три вида:

- а) законодательная метрология;
- б) фундаментальная (научная) метрология;
- в) практическая (прикладная) метрология.

Законодательная метрология – это раздел метрологии, включающий комплексы взаимосвязанных и взаимообусловленных общих правил, а также другие вопросы, нуждающиеся в регламентации и контроле со стороны государства, направленные на обеспечение единства измерений и единообразия средств измерений. Это означает, что законодательная метрология служит средством государственного регулирования метрологической деятельности посредством законов и законодательных положений, которые вводятся в практику через Государственную метрологическую службу (ГМС) и метрологические службы государственных органов управления и юридических лиц.

К области законодательной метрологии относятся испытания и утверждение типа средств измерений (СИ), государственный метрологический контроль и надзор за СИ, а также мероприятия по реальному обеспечению единства измерений. Так, одна из основных задач метрологии – это обеспечение единства измерений. Эта задача может быть решена при соблюдении двух основополагающих условий: выражение результатов измерений в единых узаконенных единицах; установление допустимых погрешностей результатов измерений и пределов, за которые они не должны выходить при заданной вероятности.

Единство измерений - это состояние измерений, при котором их результаты выражены в узаконенных единицах, а погрешности известны с заданной вероятностью и не выходят за установленные пределы. Иначе говоря, единство измерений необходимо для того, чтобы можно было сопоставлять результаты измерений, выполненных различными измерительными устройствами в разных местах и в разное время. Причем

сохранение единства измерений является важным как внутри страны, так и во взаимоотношениях между странами.

Фундаментальная и практическая метрологии появились еще в древние времена. К примеру, в Древней Руси основой системы мер были древнеегипетские единицы измерений, заимствованные в Древней Греции и Риме. Наименования единиц и их размеры соответствовали возможности осуществления измерений «подручными» способами, не прибегая к специальным устройствам. Так, на Руси единицами длины были в разное время: локоть (от сгиба локтя до конца среднего пальца руки); пядь (расстояние между концами большого и указательного пальца взрослого человека); аршин (его появление повлекло исчезновение пяди –  $\frac{1}{4}$  аршина); сажень (русская мера = 3 локтя = 152 см); косая сажень = 248 см. Указом Петра 1 русские меры длины были согласованы с английскими: дюйм («палец» = 2,54 см); английский фут = 12 дюймов = 30,48 см.

Первая метрическая система мер была введена во Франции в 1840 году. Ее значимость подчеркивал Д.И. Менделеев как средство содействия «будущему желанному сближению народов». С развитием науки и техники требовались новые измерения и новые единицы измерения, что стимулировало развитие фундаментальной и прикладной метрологии. Первоначально прототип единиц измерения искали в природе, исследуя макрообъекты и их движение. Так, секунда – часть периода обращения Земли вокруг своей оси. Постепенно поиски переместились на атомный и внутриатомный уровень.

Таким образом, метрология, как наука, динамически развивается.

Вариант 3

### **Что представляет информационно-управляющая система?**

Анализируя современное состояние систем автоматизации в Казахстане и в странах ближнего зарубежья, специалисты по автоматизации отмечают, что главной отличительной особенностью нашего рынка автоматизации является уникальное разнообразие систем и решений. Структура информационно управляющей системы (ИУС) представляется множеством пирамид. Пирамида используется для представления иерархии связей локальных технических систем. Однако если мы хотим показать свойства всей ИУС, пирамида распадается на множество частей. Острие пирамиды собирается в одну точку к центру круга управление нарушается, наиболее простая пирамида отражает иерархию технических устройств.

Это пирамида локальной автоматики, в основании пирамиды на первой ступени располагается управление технологическими процессами, на второй ступени - MES-системы, на третьей ступени - ERP-системы, четвертая ступень - аналитика и прогнозирующие системы (OLAP, CRM, B2B). Для этой пирамиды характерно, что на нижний уровень приходит информация в виде

технического задания, в MES-системах напротив, определенного технического задания нет и нет конкретизации задачи управления. Это связано с тем, что управление рассматривается с различными критериями: технологическими, техническими, социальными и другими.

Следовательно, круг управления с вложенными концентрическими кругами подсистем различных ступеней имеет иерархическое строение: (I) приборные системы, (II) локальные системы ограниченной связности, (III) большие ИУС уровня цеха (отдела), (IV) системы интеграции. На этот круг вставлены пирамиды функций предприятия, соответствующие его самостоятельным организационным подразделениям. Обычно, круг пирамид распадается на множество объектов, а любое предприятие должно обладать следующими объектами: объекты материального производства, объекты экономико-социального профиля, объекты обеспечения прошлого, объекты обеспечения будущего.

Таким образом, иерархия информационных систем по функциональным задачам представлена по принципу: уровень - исполнитель; уровень - руководитель технического бюро (цеха).

#### Вариант 4

### **Система автоматического управления**

Система автоматического управления (САУ) поддерживает или улучшает функционирование управляемого объекта. В ряде случаев вспомогательные для САУ операции (пуск, остановка, контроль, наладка и т.д.) также могут быть автоматизированы. САУ функционирует в основном в составе производственного или какого-либо другого комплекса.

История техники насчитывает много ранних примеров конструкций, обладающих всеми отличительными чертами САУ. Первой замкнутой САУ, получившей широкое техническое применение, была система автоматического регулирования с центробежным регулятором в паровой машине Уатта (1784). По мере совершенствования паровых машин, турбин и двигателей внутреннего сгорания всё более широко использовались различные механические регулирующие системы и устройства, достигшие значительного развития в конце XIX - начале XX вв.

Новый этап в автоматическом управлении характеризуется внедрением в системы регулирования и управления электронных элементов и устройств автоматики и телемеханики. Это обусловило появление высокоточных систем слежения и наведения, телеуправления и телеизмерения, системы автоматического контроля и коррекции. 50-е годы XX века ознаменовались появлением сложных систем управления производственными процессами и промышленными комплексами на базе электронных управляющих вычислительных машин.



САУ классифицируются в основном по цели управления, типу контура управления и способу передачи сигналов. Первоначально перед САУ ставились задачи поддержания определённых законов изменения во времени управляемых величин. В этом классе систем различают системы автоматического регулирования (САР), в задачу которых входит сохранение постоянными значения управляемой величины; системы программного управления, где управляемая величина изменяется по заданной программе; следящие системы, для которых программа управления заранее неизвестна.

## Вариант 5

### **Недостатки файловых систем**

Несмотря на появление файлов с произвольным доступом, стало очевидным, что файловые системы любого типа обладают некоторыми недостатками.

Во-первых, это избыточность данных. Многие приложения используют свои собственные файлы данных, и, следовательно, одна и та же единица данных повторяется в различных файлах. Например, в банке одно и то же имя клиента встречается в файлах, содержащих сведения о сберегательных, текущих счетах, о ссудах. Более того, хотя это одно и то же имя клиента, соответствующие поля в разных файлах могут называться по-разному. Кроме того, одно и то же поле в разных файлах может иметь разную длину. Вследствие такой избыточности необходимы лишние затраты на поддержание и хранение данных. Эта избыточность может порождать противоречия между разными версиями общих данных. В частности, информационные системы, использующие базы данных, позволяют избавиться от подобной избыточности, поскольку все приложения используют один и тот же набор данных. Существенная информация записывается один раз, то есть все приложения будут пользоваться согласованными данными.

Во-вторых, слабый контроль данных. В файловой системе нет централизованного контроля. Один и тот же элемент данных может иметь различные имена, терминология в различных отделах может быть разной. Например, банк может в термин счет вкладывать один смысл применительно к сбережениям и совсем другой применительно к ссудам. Разные значения одного и того же термина называют омонимами. И, наоборот, разные слова могут иметь одинаковые значения. Например, банк может говорить о владельце счета или клиенте, вкладывая в этот термин один и тот же смысл. Термины, имеющие одно и то же значение, называются синонимами. Система управления базами данных осуществляет централизованный контроль данных и помогает избежать недоразумений, порожденных омонимами и синонимами.

Далее, поскольку файловые системы являются общим хранилищем файлов, принадлежащих разным пользователям, системы управления файлами

должны обеспечивать авторизацию доступа к файлам. Это означает, что подход состоит в том, что по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю.

В-третьих, недостаточные возможности управления данными. Индексные файлы позволяют обращаться к определенной записи по ключу. Этого достаточно до тех пор, пока нам нужна отдельная запись. Если же нам нужен набор связанных данных, то такую информацию трудно или даже невозможно извлечь из файловой системы. Иными словами, файловые системы не позволяют устанавливать связь между данными разных файлов. Системы управления базами данных были специально разработаны для того, чтобы упростить связывание данных из разных файлов.

В-четвертых, большие затраты труда программиста. В файловой системе новая прикладная программа требует нового набора файлов. Даже если существующий файл содержит необходимую информацию, приложению требовался еще какой-либо набор данных. В результате программисту надо писать еще одну программу, а именно между данными и программами существовала жесткая зависимость. Базы данных позволили разделить программы и данные, так что программа в некотором смысле независима от деталей определения данных.

Таким образом, названные недостатки послужили тем толчком, который заставил разработчиков информационных систем предложить новый подход к управлению информацией. Этот подход был реализован в рамках новых программных систем, названных впоследствии Системами Управления Базами Данных (СУБД), а сами хранилища информации, которые работали под управлением данных систем, назывались базами или банками данных (БД и БнД).

## Вариант 6

### **Информационные системы, использующие базы данных**

Терминология в системе управления базами данных (СУБД), да и сами термины «база данных» и «банк данных» частично заимствованы из финансовой деятельности. Это заимствование не случайно и объясняется тем, что работа с информацией и работа с денежными массами во многом схожи, поскольку и там и там отсутствует персонификация объекта обработки: две банкноты достоинством в сто рублей столь же неотличимы и взаимозаменяемы, как два одинаковых байта (естественно, за исключением серийных номеров). Вы можете положить деньги на некоторый счет и предоставить возможность вашим родственникам или коллегам использовать их для иных целей. Вы можете поручить банку оплачивать ваши расходы с вашего счета или получить их наличными в другом банке, и это будут уже

другие денежные купюры, но их ценность будет эквивалентна той, которую вы имели, когда клали их на ваш счет.

База данных (БД) – упорядоченный набор хранимых данных, связанных общей темой или назначением. Основное назначение базы данных – быстрый поиск содержащейся в ней информации. Система управления базами данных (СУБД) – программное обеспечение, осуществляющее управление базой данных. Это означает, что понятие базы данных шире, чем просто набор данных. Кроме собственно базы данных имеется набор прикладных программ, которые работают с этими данными, обрабатывая их обычным способом, а также соответствующее оборудование и люди.

В общем случае с одной базой данных могут работать множество различных приложений. Например, если база данных моделирует некоторое предприятие, то для работы с ней может быть создано приложение, которое обслуживает подсистему учета кадров, другое приложение может быть посвящено работе подсистемы расчета заработной платы сотрудников, третье приложение работает как подсистема складского учета, четвертое приложение посвящено планированию производственного процесса. При рассмотрении приложений, работающих с одной базой данных, предполагается, что они могут работать параллельно и независимо друг от друга, и именно СУБД призвана обеспечить работу множества приложений с единой базой данных таким образом, чтобы каждое из них выполнялось корректно, но учитывало все изменения в базе данных, вносимые другими приложениями.

Информационная система, использующая базы данных, точнее система баз данных состоит из четырех компонентов: оборудования, программного обеспечения (СУБД), данных (БД) и людей (пользователи, администрация, обслуживающий персонал).

В удачно разработанной и функционирующей системе все четыре компонента взаимодействуют, образуя единую систему, выполняющую нужные задачи: обслуживающий персонал, советуясь с пользователями, определяет необходимые данные и создает структуру базы данных, отвечающую потребностям пользователей. Затем структура базы данных сообщается СУБД через словарь данных. Пользователи вводят в систему данные, следуя определенным процедурам (инструкциям). Хранение введенных данных обеспечивается оборудованием.

Итак, прикладные программы, обслуживающие доступ к базе данных, разрабатываются программистами, а пользователи запускают их на компьютерах. Эти программы выдают информацию, которая нужна руководству компании и ее клиентам.

Вариант 7

## **Базы данных на больших ЭВМ**

Информационные системы, использующие базы данных, позволили преодолеть недостатки файловых систем. Одна из основных целей систем баз данных - обеспечение независимости данных, то есть независимости приложений от изменений в структуре хранения и стратегии доступа. В базе данных может поддерживаться целостность данных, то есть их точность и непротиворечивость. Стремительное развитие вычислительной техники, изменение ее принципиальной роли в жизни общества повлияло также и на развитие технологии баз данных. Можно выделить четыре этапа в развитии данного направления в обработке данных. Однако необходимо заметить, что все же нет жестких временных ограничений в этих этапах.

Первый этап развития системы управления базами данных (СУБД) связан с организацией баз данных на больших машинах. Базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи, запускаемые в основном в пакетном режиме. Интерактивный режим доступа обеспечивался с помощью консольных терминалов, которые не обладали собственными вычислительными ресурсами и служили только устройствами ввода-вывода для центральной ЭВМ. Программы доступа к базе данных писались на различных языках и запускались как обычные числовые программы. А именно, мощные операционные системы обеспечивали возможность условно параллельного выполнения всего множества задач.

Особенности этого этапа развития выражаются в следующем:

- все СУБД базируются на мощных мультипрограммных операционных системах, поэтому в основном поддерживается работа с централизованной базой данных в режиме распределенного доступа;
- функции управления распределением ресурсов в основном осуществляются операционной системой;
- поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным;
- значительная роль отводится администрированию данных;
- проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и создается первая система (System R), реализующая идеологию реляционной модели данных;
- результаты научных исследований открыто обсуждаются в печати, идет мощный поток общедоступных публикаций, касающихся всех аспектов теории и практики баз данных, и результаты теоретических исследований активно внедряются в коммерческие СУБД;
- появляются первые языки высокого уровня для работы с реляционной моделью данных.

Вариант 8

**Эпоха персональных компьютеров**

Персональные компьютеры стремительно перевернули представление о месте и роли вычислительной техники в жизни общества. Появилось множество программ, предназначенных для работы неподготовленных пользователей. Системные программисты были отодвинуты на второй план. И, конечно, это сказалось и на работе с базами данных. Так, появились программы, которые назывались системами управления базами данных и позволяли хранить значительные объемы информации, они имели удобный интерфейс для заполнения данных, встроенные средства для генерации различных отчетов. Эти программы позволяли автоматизировать многие учетные функции, которые раньше велись вручную.

Постоянное снижение цен на персональные компьютеры сделало их доступными не только для организаций и фирм, но и для отдельных пользователей. А именно, компьютеры стали инструментом для ведения документации и собственных учетных функций. Это все сыграло как положительную, так и отрицательную роль в области развития баз данных. В частности, кажущаяся простота и доступность персональных компьютеров и их программного обеспечения породила множество дилетантов. Однако доступность персональных компьютеров заставила пользователей из многих областей знаний, которые ранее не применяли вычислительную технику в своей деятельности, обратиться к ним. И спрос на развитые удобные программы обработки данных заставлял поставщиков программного обеспечения поставлять все новые системы, которые принято называть настольными (desktop) системами управления базами данных (СУБД).

Этот этап имел свои особенности. Во-первых, все СУБД были рассчитаны на создание баз данных, в основном, с монопольным доступом и это понятно. Компьютер персональный, точнее он не был подсоединен к сети, и база данных на нем создавалась для работы одного пользователя. В редких случаях предполагалась последовательная работа нескольких пользователей, например, сначала оператор, который вводил бухгалтерские документы, а потом главбух, который определял проводки, соответствующие первичным документам.

Во-вторых, большинство СУБД имели развитый и удобный пользовательский интерфейс. В большинстве существовал интерактивный режим работы с базой данных, как в рамках ее описания, так и в рамках проектирования запросов. Кроме того, большинство СУБД предлагали развитый и удобный инструментарий для разработки готовых приложений без программирования. Инструментальная среда состояла из готовых элементов приложения в виде шаблонов экранных форм, отчетов, этикеток, графических конструкторов запросов, которые достаточно просто могли быть собраны в единый комплекс.

В-третьих, во всех настольных СУБД поддерживался только внешний уровень представления реляционной модели, то есть только внешний табличный вид структур данных.

В-четвертых, при наличии высокоуровневых языков манипулирования данными типа реляционной алгебры и SQL в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.

В-пятых, в настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных. Эти функции должны были выполнять приложения, однако скудость средств разработки приложений иногда не позволяла это сделать, и в этом случае эти функции должны были выполняться пользователем, требуя от него дополнительного контроля при вводе и изменении информации, хранящейся в базе данных.

В-шестых, наличие монопольного режима работы фактически привело к вырождению функций администрирования и в связи с этим - к отсутствию инструментальных средств администрирования баз данных.

И, наконец, последняя и в настоящий момент, весьма положительная особенность - это сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД. Вполне работоспособные приложения, разработанные, например, на Clipper, работали на PC 286.

## Вариант 9

### **Распределенные базы данных**

Хорошо известно, что история развивается по спирали, поэтому после процесса «персонализации» начался обратный процесс - интеграция. Множится количество локальных сетей, все больше информации передается между компьютерами, остро встает задача согласованности данных, хранящихся и обрабатываемых в разных местах, но логически друг с другом связанных, возникают задачи, связанные с параллельной обработкой транзакций-последовательностей операций над базой данных, переводящих ее из одного непротиворечивого состояния в другое непротиворечивое состояние. Это означает, что успешное решение этих задач приводит к появлению распределенных баз данных, сохраняющих все преимущества настольных систем управления базами данных (СУБД) и в то же время позволяющих организовать параллельную обработку информации и поддержку целостности баз данных (БД).

Особенности данного этапа характеризуются рядом факторов. В частности, практически все современные СУБД обеспечивают поддержку полной реляционной модели. Большинство современных СУБД рассчитаны на многоплатформенную архитектуру, точнее они могут работать на компьютерах с разной архитектурой и под разными операционными системами, при этом для пользователей доступ к данным, управляемым СУБД на разных платформах, практически неразличим.

Необходимость поддержки многопользовательской работы с базой данных и возможность децентрализованного хранения данных потребовали

развития средств администрирования баз данных с реализацией общей концепции средств защиты данных. Потребность в новых реализациях вызвала создание серьезных теоретических трудов по оптимизации реализаций распределенных баз данных и работе с распределенными транзакциями и запросами с внедрением полученных результатов в коммерческие СУБД. Все современные СУБД имеют средства подключения клиентских приложений, разработанных с использованием настольных СУБД, и средства экспорта данных из форматов настольных СУБД второго этапа развития.

## Вариант 10

### **Предметная область информационной системы**

Каждая информационная система в зависимости от ее назначения имеет дело с той или иной частью реального мира, который принято называть предметной областью. Точнее, предметная область - некоторая совокупность реальных объектов, которые представляют интерес для ее пользователей. Существует этап, предшествующий этапу проектирования базы данных. Модель этого этапа должна выражать информацию о предметной области в виде, независимом от базы данных, используемой системой управления. Итак, на первом этапе проектирования необходимо выполнить системный анализ предметной области. С точки зрения проектирования базы данных в рамках системного анализа, необходимо провести подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами. Желательно, чтобы данное описание позволяло корректно определить все взаимосвязи между объектами предметной области. В частности, существуют два подхода к выбору состава и структуры предметной области.

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

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

Конструирование предметной базы данных в некотором смысле кажется гораздо более заманчивым, однако трудность всеобщего охвата предметной области с невозможностью конкретизации потребностей пользователей может привести к избыточно сложной схеме базы данных, которая для конкретных задач будет неэффективной. Чаще всего на практике рекомендуется использовать некоторый компромиссный вариант, который, с одной стороны, ориентирован на конкретные задачи или функциональные потребности пользователей, а с другой стороны, учитывает возможность наращивания новых приложений.

Таким образом, системный анализ должен заканчиваться подробным описанием информации об объектах предметной области, которая требуется для решения конкретных задач и которая должна храниться в базе данных; формулировкой конкретных задач, которые будут решаться с использованием данной базы; описанием входных документов, которые служат основанием для заполнения данными базы данных; кратким описанием алгоритмов решения задач; описанием выходных документов, которые должны генерироваться в системе.

## Вариант 11

### **Разработка информационной системы**

Пусть требуется разработать информационную систему для автоматизации учета получения и выдачи книг в библиотеке. Система должна предусматривать режимы ведения системного каталога, отражающего перечень областей знаний, по которым имеются книги в библиотеке. Внутри библиотеки области знаний в систематическом каталоге могут иметь уникальный внутренний номер и полное наименование. В частности, каждая книга может содержать сведения из нескольких областей знаний. Каждая книга в библиотеке может присутствовать в нескольких экземплярах. К примеру, каждая книга, хранящаяся в библиотеке, характеризуется следующими параметрами: уникальный шифр; название; фамилии авторов (могут отсутствовать); место издания (город); издательство; год издания; количество страниц; стоимость книги; количество экземпляров книги в библиотеке. Книги могут иметь одинаковые названия, но они различаются по своему уникальному шифру (ISBN).

В библиотеке ведется картотека читателей. Так, на каждого читателя в картотеку заносятся следующие сведения: фамилия, имя, отчество; домашний адрес; телефон (рабочий и домашний); дата рождения. Каждому читателю присваивается уникальный номер читательского билета. Каждый читатель может одновременно держать на руках не более 5 книг. Читатель не должен одновременно держать более одного экземпляра книги одного названия. Каждая книга в библиотеке может присутствовать в нескольких экземплярах. Это означает, что каждый экземпляр имеет следующие характеристики:



уникальный инвентарный номер; шифр книги, который совпадает с уникальным шифром из описания книг; место размещения в библиотеке. В случае выдачи экземпляра книги читателю в библиотеке хранится специальный вкладыш, в котором должны быть записаны следующие сведения: номер билета читателя, который взял книгу; дата выдачи книги; дата возврата.

Предусматриваются следующие ограничения на информацию в системе: книга может не иметь ни одного автора; в библиотеке должны быть записаны читатели не моложе 17 лет; в библиотеке присутствуют книги, изданные начиная с 1960 по текущий год; каждый читатель может держать на руках не более 5 книг; каждый читатель при регистрации в библиотеке должен дать телефон для связи; каждая область знаний может содержать ссылки на множество книг, и каждая книга может относиться к различным областям знаний. С данной информационной системой должны работать следующие группы пользователей: библиотекари; читатели; администрация библиотеки. При работе с системой библиотекарь должен иметь возможность решать следующие задачи: принимать новые книги и регистрировать их в библиотеке; проводить каталогизацию книг; проводить списание старых и не пользующихся спросом книг; вести учет выданных книг читателям; проводить списание утерянных читателем книг; проводить закрытие абонемента читателя. Читатель должен иметь возможность решать следующие задачи: просматривать системный каталог; по выбранной области знаний получить полный перечень книг, находящихся в библиотеке; для выбранного автора получить список книг, которые числятся в библиотеке.

Администрация библиотеки должна иметь возможность получать сведения о должниках-читателях библиотеки; сведения о книгах, которые не являются популярными, т. е. ни один экземпляр которых не находится на руках у читателей; сведения о стоимости конкретной книги, для того, чтобы установить возможность возмещения стоимости утерянной книги или возможность замены ее другой книгой; сведения о наиболее популярных книгах, то есть таких, все экземпляры которых находятся на руках у читателей.

Таким образом, данный пример показывает, что перед началом разработки необходимо иметь точное представление о том, что же должно выполняться в нашей системе, какие пользователи в ней будут работать, какие задачи будет решать каждый пользователь.

## Вариант 12

### **Архитектура базы данных**

Важным аспектом развития методов доступа к данным стала идея отделения логической структуры и манипуляции данными, как они понимаются пользователями, от физического представления, требуемого

компьютерным оборудованием. В процессе научных исследований, посвященных тому, как именно должна быть устроена система управления базами данных (СУБД), предлагались различные способы реализации. Например, самым жизнеспособным из них оказалась предложенная американским комитетом по стандартизации ANSI (American National Standards Institute) трехуровневая система организации базы данных, то есть стандартная структура систем баз данных, состоящая из внешнего, концептуального и внутреннего уровней.

Уровень внешних моделей - самый верхний уровень, где каждая модель имеет свое «видение» данных. Этот уровень определяет точку зрения на базу данных отдельных приложений. Каждое приложение видит и обрабатывает только те данные, которые необходимы именно этому приложению. Например, система распределения работ использует сведения о квалификации сотрудника, но ее не интересуют сведения об окладе, домашнем адресе и телефоне сотрудника, и наоборот, именно эти сведения используются в подсистеме отдела кадров. Внешний уровень - структурный уровень базы данных, определяющий пользовательские представления базы данных.

Концептуальный уровень - центральное управляющее звено, здесь база данных представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями, работающими с данной базой данных. Фактически концептуальный уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась база данных. Как любая модель, концептуальная модель отражает только существенные, с точки зрения обработки, особенности объектов реального мира. Концептуальный уровень – структурный уровень базы данных, определяющий логическую схему базы данных. Концептуальное проектирование базы данных, выполняемой на концептуальном уровне, включает анализ информационных потребностей пользователей и определение нужных им элементов данных.

Физический уровень - собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации. Внутренний уровень –структурный уровень базы данных, определяющий физический вид базы данных. Он связан со способом физического хранения данных: дисководы, физические адреса, индексы, указатели и т.д. Так, ни один пользователь (как пользователь) не касается этого уровня.

Таким образом, если внешний уровень связан с «частными» представлениями пользователей, то концептуальный уровень можно представить себе определяющим обобщенное представление пользователей. Это представление ближе к данным «как они есть», чем какими их видят пользователи. Это означает, что может быть много «внешних представлений», каждое из которых состоит из представления части базы данных, и может быть единственным «концептуальным представлением», состоящим из абстрактного представления базы данных в целом. Также есть единственное

«внутреннее» представление, отражающее всю базу данных как действительно хранимую.

## Вариант 13

### **Концептуальные модели данных**

Одними из основополагающих в концепции баз данных являются обобщенные категории «данные» и «модель данных». Понятие «данные» в концепции баз данных - это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы. Это означает, что данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознает их смысловое содержание. Поэтому центральным понятием в области баз данных является понятие модели. Не существует однозначного определения этого термина, у разных авторов эта абстракция определяется с некоторыми различиями, но, тем не менее, можно выделить общее в этих определениях. Иными словами, база данных - это совокупность данных, структурированных определенным образом. Структура базы данных воплощает структуру предметной области. Модель - представление реальности, отражающее только избранные детали. В контексте баз данных модель данных – методы структурирования данных. В соответствии с рассмотренной ранее трехуровневой архитектурой мы сталкиваемся с понятием модели данных по отношению к каждому уровню. К примеру, ядром любой системы баз данных является концептуальная модель. На концептуальном уровне выполняется концептуальное проектирование, то есть создание концептуальной схемы базы данных. На этом этапе создаются подробные модели пользовательских представлений данных, затем они интегрируются в концептуальную модель, фиксирующую все элементы данных, которые будет содержать база данных.

Концептуальная модель разрабатывается после словесного описания предметной области. Эта модель должна включать такое формализованное описание предметной области, которое легко будет «читаться» не только специалистами по базам данных и оно не должно быть привязано к конкретной системе управления базами данных (СУБД). Одним словом, выбор системы управления базой данных - это отдельная задача, для корректного ее решения необходимо иметь проект, который не привязан ни к какой конкретной СУБД. Инфологическое проектирование, прежде всего, связано с попыткой представления семантики (смысла) предметной области в модели базы данных. Реляционная модель данных в силу своей простоты и лаконичности не позволяет отобразить семантику, то есть смысл предметной области.

Проблема представления семантики давно интересовала разработчиков. В частности, в семидесятых годах прошлого века было предложено несколько моделей данных, названных семантическими моделями. Семантическое моделирование представляет собой моделирование структуры данных, опираясь на смысл этих данных. К примеру, к таким моделям можно отнести семантическую модель данных, предложенную Хаммером (Hammer) и Мак-Леоном (McLeon) в 1981 году, функциональную модель данных Шипмана (Shipman), также созданную в 1981 году, модель «сущность—связь», предложенную Питером Пин-Шэн Ченом (Chen) в 1976 году, и ряд других моделей.

Итак, у всех моделей были свои положительные и отрицательные стороны, но испытание временем выдержала только последняя. В дальнейшем многими авторами были разработаны свои варианты подобных моделей (нотация Мартина, нотация IDEF1X, нотация Баркера и др.).

#### Вариант 14

### **Определения концептуальных моделей данных**

Как любая модель, концептуальная модель имеет несколько базовых понятий, которые образуют исходные кирпичики, из которых строятся уже более сложные объекты по заранее определенным правилам. Эта модель в наибольшей степени согласуется с концепцией объектно-ориентированного проектирования, которая в настоящий момент, несомненно, является базовой для разработки сложных программных систем. Главными элементами концептуальной модели являются объекты и отношения. Объекты— вещи, которые пользователи считают важными в моделируемой части реальности. Все множество предметной области разбивается на группы объектов, однородных по структуре и поведению, называемых типами объектов. Объектное множество — множество вещей одного типа. Объект-элемент— конкретный элемент объектного множества. Объекты часто представляют в виде существительных, а отношения - в виде глаголов. Также для обозначения объекта может использоваться термин «сущность». Имя объектного множества пишется заглавными буквами в единственном числе. Например, ЧЕЛОВЕК - имя объектного множества, представляющего людей, строчными буквами «человек» обозначается элемент из объектного множества.

Объектные множества бывают лексическими и абстрактными. Лексическое объектное множество — объектное множество, состоящее из элементов, которые можно напечатать. Абстрактное объектное множество - объектное множество, состоящее из элементов, которые нельзя напечатать. В компьютерной реализации концептуальной модели элементы лексических объектов будут представлены в виде строк символов. В частности, элементы абстрактных множеств будут представлены внутренними номерами, не имеющими смысла вне системы. Внутренний номер называют

«идентификатором объекта» или суррогатным ключом, так как он представляет и однозначно определяет абстрактный объект-элемент реального мира. Суррогатный ключ - «идентификатор» абстрактного объекта-элемента в компьютерной системе, вне системы не имеет смысла.

Объекты могут быть базовыми (основными) и подчиненными (дочерними). Например, читатель библиотеки – базовая сущность, а абонемент этого читателя – подчиненная, которая зависит от наличия соответствующего читателя. Некоторые объектные множества содержатся внутри объектных множеств. Конкретизация – объектное множество, являющееся подмножеством другого объектного множества. Обобщение – объектное множество, являющееся надмножеством другого объектного множества (содержащее его).

Между объектами могут быть установлены связи- бинарные ассоциации, показывающие, каким образом объекты соотносятся или взаимодействуют между собой. Связь может существовать между двумя разными объектами или между объектом и им же самим (рекурсивная связь). Она показывает, как связаны экземпляры объектных множеств между собой. Связь между элементами двух объектных множеств называется отношением. Отношение само по себе является объектным множеством, состоящим из пар объектов-элементов, взятых из двух множеств, которые соединяет отношение.

Итак, отношение, рассматриваемое как объектное множество, называется составным объектным множеством. Составным объектным множествам можно давать имена и включать их в отношения, как и обычные объектные множества.

## Вариант 15

### **Графическое представление концептуальной модели**

После словесного описания предметной области, на втором этапе проектирования, необходимо разработать концептуальную модель базы данных. Концептуальная модель должна включать такое формализованное описание предметной области, которое легко будет «читаться» не только специалистами по базам данных. В частности, это описание должно быть настолько емким, чтобы можно было оценить глубину и корректность проработки проекта базы данных, и, конечно, оно не должно быть привязано к конкретной системе управления базами данных (СУБД). Иными словами, выбор СУБД - это отдельная задача, для корректного ее решения необходимо иметь проект, который не привязан ни к какой конкретной СУБД.

При концептуальном моделировании баз данных используется ER-модель, представляющую базу данных в виде сущностей и связей между ними. В настоящий момент не существует единой общепринятой системы обозначений для ER-модели, используются разные графические нотации, но разобравшись в одной, можно легко понять и другие. В частности, одно из

общепринятых графических обозначений объектного множества - прямоугольник, тогда объекты-элементы обозначаются в виде точек. Другое обозначение объекта - прямоугольник, в верхней части которого записано имя сущности, а ниже перечисляются атрибуты, причем ключевые атрибуты помечаются, например, подчеркиванием или специальным шрифтом.

Связи между объектными множествами обозначаются линиями, посередине которых пишут название связи. Иногда для наглядности название связи размещают в ромбе посередине этой линии. В разных нотациях мощность связи изображается по-разному. Обязательность связи тоже обозначается по-разному. Например, необязательность связи можно обозначить пустым кружочком на конце связи, а обязательность перпендикулярной линией, перечеркивающей связь. Графическая интерпретация связи позволяет сразу прочесть смысл взаимосвязи между объектами, она наглядна и легко интерпретируема. В качестве примера приведем ER-диаграмму, отражающую связи между объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ, где связь - руководство дипломными проектами: каждый студент имеет только одного руководителя, но один и тот же преподаватель может руководить множеством студентов-дипломников – «связь один - ко-многим».

Итак, в результате построения модели предметной области в виде набора объектов и связей получаем связный граф. В полученном графе необходимо избегать циклических связей - они выявляют некорректность модели.

## Вариант 16

### **Задачи концептуального проектирования**

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на программное обеспечение с позиции будущих пользователей базы данных. В качестве примера спроектируем концептуальную модель системы, предназначенной для хранения информации о книгах и областях знаний, представленных в библиотеке. Выделим основные объектные множества. В частности, существует объектное множество КНИГИ. Каждая книга имеет уникальный шифр, который является ее ключом, и ряд атрибутов, которые взяты из описания предметной области. Множество экземпляров множества определяет множество книг, которые хранятся в библиотеке. Каждый экземпляр множества КНИГИ соответствует не конкретной книге, стоящей на полке, а описанию некоторой книги, которое дается обычно в предметном каталоге библиотеки. Каждая книга может присутствовать в нескольких экземплярах, и это как раз те конкретные книги, которые стоят на полках библиотеки. Для того чтобы отразить это, мы должны ввести объектное множество ЭКЗЕМПЛЯРЫ, которое будет содержать описания всех

экземпляров книг, хранящихся в библиотеке. Это означает, что каждый экземпляр сущности ЭКЗЕМПЛЯРЫ соответствует конкретной книге на полке.

Каждый экземпляр имеет уникальный инвентарный номер, однозначно определяющий конкретную книгу. Кроме того, каждый экземпляр книги может находиться либо в библиотеке, либо на руках у некоторого читателя, и в последнем случае для данного экземпляра указываются дополнительно дата взятия книги читателем и дата предполагаемого возврата книги. А именно, связи бывают обязательные и факультативные. Если объект одного типа по необходимости оказывается связанным с объектом другого типа, то между этими типами объектов существует обязательная связь. Иначе связь является факультативной.

Например, между сущностями КНИГИ и ЭКЗЕМПЛЯРЫ есть связь «один - ко-многим» (1:M), обязательная с двух сторон. Чем определяется данный тип связи? Иными словами, мы можем предположить, что каждая книга может присутствовать в библиотеке в нескольких экземплярах, поэтому связь «один - ко-многим». При этом если в библиотеке нет ни одного экземпляра данной книги, то мы не будем хранить ее описание, поэтому если книга описана в сущности КНИГИ, то, по крайней мере, один экземпляр этой книги присутствует в библиотеке. Это означает, что со стороны книги связь обязательная. Что касается сущности ЭКЗЕМПЛЯРЫ, то не может существовать в библиотеке ни одного экземпляра, который бы не относился к конкретной книге, поэтому и со стороны ЭКЗЕМПЛЯРЫ связь тоже обязательная.

Теперь нам необходимо определить, как в нашей системе будет представлен читатель. Естественно предложить ввести для этого объектное множество ЧИТАТЕЛИ, каждый экземпляр которой будет соответствовать конкретному читателю. В библиотеке каждому читателю присваивается уникальный номер читательского билета, который будет однозначно идентифицировать нашего читателя. Номер читательского билета будет ключевым атрибутом сущности ЧИТАТЕЛИ. Точнее, в сущности ЧИТАТЕЛИ должны присутствовать дополнительные атрибуты, которые требуются для решения поставленных задач. Например, этими атрибутами будут: «Фамилия Имя Отчество», «Адрес читателя», «Телефон домашний» и «Телефон рабочий».

Таким образом, из описания предметной области мы знаем, что каждый читатель может держать на руках несколько экземпляров книг. Для отражения этой ситуации нам надо провести связь между сущностями ЧИТАТЕЛИ и ЭКЗЕМПЛЯРЫ, потому что читатель берет из библиотеки конкретный экземпляр конкретной книги, а не просто книгу.

Вариант 17

## **Иерархическая модель данных**

Первая информационная система, использующая базы данных, появившаяся в середине 60-х годов XX века, была основана на иерархической модели. Иерархическая модель - модель данных, в которой связи между данными имеют вид иерархий. Исторически эти модели появились раньше, и в настоящий момент они используются реже, чем современные реляционные модели данных. Однако до сих пор существуют системы, работающие на основе этих моделей, а одна из концепций развития объектно-ориентированных баз данных предполагает объединение принципов сетевой модели с концепцией реляционной. Появление иерархической модели связано с тем, что в реальном мире очень многие связи соответствуют иерархии, когда один объект выступает как родительский, а с ним может быть связано множество подчиненных объектов. Это означает, что иерархия проста и естественна в отображении взаимосвязи между классами объектов.

Основными информационными единицами в иерархической модели являются: база данных (БД), сегмент и поле. Поле данных определяется как минимальная, неделимая единица данных, доступная пользователю с помощью системы управления базами данных (СУБД). Сегмент в терминологии Американской Ассоциации по базам данных DBTG (Data Base Task Group) называется записью, при этом в рамках иерархической модели определяются два понятия: тип сегмента или тип записи и экземпляр сегмента или экземпляр записи. Тип сегмента - это поименованная совокупность типов элементов данных, в него входящих. Экземпляр сегмента образуется из конкретных значений полей или элементов данных, в него входящих. Каждый тип сегмента в рамках иерархической модели образует некоторый набор однородных записей. Для возможности различия отдельных записей в данном наборе каждый тип сегмента должен иметь ключ или набор ключевых атрибутов (полей, элементов данных). Ключом называется набор элементов данных, однозначно идентифицирующих экземпляр сегмента.

В иерархической модели сегменты объединяются в ориентированный древовидный граф. При этом полагают, что направленные ребра графа отражают иерархические связи между сегментами: каждому экземпляру сегмента, стоящему выше по иерархии и соединенному с данным типом сегмента, соответствует несколько (множество) экземпляров данного (подчиненного) типа сегмента. Тип сегмента, находящийся на более высоком уровне иерархии, называется логически исходным по отношению к типам сегментов, соединенных с данными направленными иерархическими ребрами, которые в свою очередь называются логически подчиненными по отношению к этому типу сегмента. Иногда исходные сегменты называют сегментами-предками, а подчиненные сегменты называют сегментами-потомками.

Таким образом, понятие схемы базы данных определяется на концептуальном уровне в качестве иерархической модели. Схема иерархической базы данных представляет собой совокупность отдельных деревьев, каждое дерево в рамках модели называется физической базой



данных. Каждая физическая база данных удовлетворяет определенным иерархическим ограничениям.

## Вариант 18

### Сетевая модель данных

У иерархической модели есть недостаток, так как не все отношения можно представить в виде иерархии. Сеть – отношения между данными, когда каждая запись может быть подчинена записям более чем из одного файла. В связи с очевидной необходимостью обрабатывать такие отношения, в конце 60-годов XX века появились сетевые системы управления базами данных. Как и в иерархических системах, в сетевых системах баз данных для связывания данных использовались предопределенные физические указатели. В иерархии у каждого потомка (подчиненной записи в файле) может быть только один предок (подчиняющая запись в иерархии). Иными словами, сетевые модели поддерживали более сложные отношения между записями из разных файлов.

Базовыми объектами модели являются: элемент данных, агрегат данных, запись и набор данных. Элемент данных - то же, что и в иерархической модели, то есть минимальная информационная единица, доступная пользователю с использованием системы управления базами данных. Агрегат данных соответствует следующему уровню обобщения в модели. В модели определены агрегаты двух типов: агрегат типа «вектор» и агрегат типа «повторяющаяся группа». Агрегат данных имеет имя, и в системе допустимо обращение к агрегату по имени.

Записью называется совокупность агрегатов или элементов данных, моделирующая некоторый класс объектов реального мира. Понятие «запись» соответствует понятию «сегмент» в иерархической модели. Для записи так же, как и для сегмента, вводятся понятия «тип записи» и «экземпляр записи».

Следующим базовым понятием в сетевой модели является понятие «набор». Набором называется двухуровневый граф, связывающий отношениями «один - ко-многим» два типа записи. Точнее, набор фактически отражает иерархическую связь между двумя типами записей. Родительский тип записи в данном наборе называется владельцем набора, а дочерний тип записи — членом того же набора. Для любых двух типов записей может быть задано любое количество наборов, которые их связывают. Фактически наличие подобных возможностей позволяет промоделировать отношение «многие - ко-многим» между двумя объектами реального мира, что выгодно отличает сетевую модель от иерархической.

В рамках набора возможен последовательный просмотр экземпляров членов набора, связанных с одним экземпляром владельца набора. Между двумя типами записей может быть определено любое количество наборов: например, можно построить два взаимосвязанных набора. Существенным

ограничением набора является то, что один и тот же тип записи не может быть одновременно владельцем и членом набора.

Таким образом, использование физических указателей позволяет извлекать данные, связанные определенными отношениями. Однако эти отношения должны быть определены до запуска системы. Извлечь данные на основе других отношений было сложно или даже невозможно.

## Вариант 19

### **Переход к реляционной модели**

Реляционная модель основывается на математических принципах, вытекающих непосредственно из теории множеств и логики предикатов. Эти принципы впервые были применены в области моделирования данных доктором Е.Ф.Коддом (в то время работавшим в IBM). Так, в 1970 году доктор опубликовал революционную по содержанию статью, которая всерьез поколебала устоявшиеся представления о базах данных. Он выдвинул идею, что данные нужно связывать в соответствии с их внутренними логическими взаимоотношениями, а не физическими указателями. Таким образом, пользователи смогут комбинировать данные из разных источников, если логическая информация, необходимая для такого комбинирования, присутствует в исходных данных. Это открыло новые возможности для информационно-управляющих систем, поскольку запросы к базам данных теперь не были ограничены физическими указателями. Информационные системы, использующие базы данных, которые поддерживают извлечение данных на основе логических связей, позволяют легко получить ответы на множество вопросов. В частности, в своей статье Кодд предложил простую модель данных, согласно которой все данные сведены в таблицы, состоящие из строк и столбцов. Эти таблицы получили название реляций, а модель стала называться реляционной.

Реляционная модель определяет способ представления данных (структуру данных), методы защиты данных (целостность данных), а также операции, выполняемые с данными (манипулирование данными). Распространено заблуждение, что реляционная модель названа так потому, что она определяет отношения между таблицами. На самом деле название этой модели происходит от отношений (реляций), лежащих в ее основе. В рамках реляционной модели данные представлены в виде отношений на концептуальном уровне, однако при этом не дается никаких указаний, каким образом данные будут реализованы на физическом уровне.

Рассматривая данные с концептуальной, а не физической точки зрения, Кодд предложил еще одну революционную идею. В реляционных системах баз данных целые файлы данных могут обрабатываться одной командой, тогда как в традиционных системах за один раз обрабатывается только одна запись. Метод разделения концептуального и логического уровней произвел

переворот в области программирования баз данных. Подход Кодда чрезвычайно повысил эффективность программирования в базах данных. Ранее программирование баз данных сводилось в основном к написанию программного кода для физического управления устройствами, предназначенными для хранения данных. А именно, логический подход к данным сделал также возможным создание языков запросов, более доступных для пользователей, не являющихся специалистами по компьютерным технологиям. Это означает, что очень трудно создать язык, которым могли бы пользоваться все, независимо от опыта работы с компьютером, однако реляционные языки запросов сделали базы данных доступными для более широкого круга пользователей, чем раньше.

Таким образом, публикация работ Кодда в начале семидесятых годов прошлого века вызвала взрыв активности, как среди ученых, так и среди разработчиков коммерческих систем по созданию реляционной системы управления базами данных. Сегодня реляционные базы данных рассматриваются как стандарт для современных коммерческих систем работы с данными.

## Вариант 20

### **Реляционные таблицы и ключи**

Реляционная модель основана на понятии «отношения» (Relationship), она наиболее распространена в настоящее время, хотя наряду с общепризнанными достоинствами обладает и рядом недостатков. Эти преимущества привели к тому, что уже в середине 80-х годов XX века реляционные СУБД (системы управления базами данных) практически вытеснили с мирового рынка ранние СУБД. Реляционная модель данных – модель данных, представляющая данные в виде таблиц или реляций. Реляция – двумерная таблица, содержащая строки и столбцы данных. Так, в модели Кодда для работы с данными в таблице предлагается пользоваться двумя языками: реляционной алгеброй и реляционным исчислением. Математическое название таблицы – relation (отношение).

Понятие таблицы - это основа реляционной модели. База данных - это набор таблиц. Это означает, что таблица имеет отношение только к представлению данных. Никакой связи с физической реализацией записей на диске или в памяти она не имеет. В реляционной СУБД выполняются как минимум два условия: данные воспринимаются пользователем как таблицы; в распоряжении пользователя имеются операторы, которые генерируют новые таблицы из старых. Здесь важно, что результат операции над таблицей является тоже таблицей (реляционное свойство замкнутости), то есть результатом операции является объект того же рода, что и объект, над которым проводится операция.

В основе реляционной модели лежат следующие понятия: *тип данных*, *атрибут*, *домен*, *кортеж*, *первичный ключ* и *отношение*. Понятие «тип данных» в реляционной модели данных полностью совпадает с этим понятием в языках программирования. В частности, выделяют символьный, числовой, денежный, битовый и календарный типы. Для хранения информации используется *атрибут*. Атрибут в реляционной модели имеет тот же смысл, что и свойство объекта в ER-модели. Атрибут имеет имя и тип. Структура отношения состоит из набора атрибутов и их типов. Название атрибута – имя атрибута. Количество атрибутов реляции называется *степенью (рангом) реляции*. Порядок атрибутов считается несущественным. Отсюда следует, что никакие два атрибута реляции не могут иметь одинаковые имена.

Набор всевозможных значений, которые может принимать атрибут, называется *доменом* (областью) атрибута. Домен играет огромную роль в реляционной модели. Он реализует функцию справочника в обыденном понятии. Применение домена позволяет избежать многих ошибок сравнения. Например, при разном вводе одного и того же понятия (к примеру, в случае ошибки ввода). Вместе с тем домен выполняет функцию контроля и классификации ввода и позволяет выбрать из имеющихся позиций необходимые значения. Все элементы домена относятся к одному типу данных и отвечают какому-либо логическому условию. Элементом домена может быть число, символьная строка, дата, но не сложная структура типа массива, списка и т.п. Две области атрибутов совпадают только в том случае, если они имеют один и тот же смысл. Два атрибута одной и той же области не обязательно имеют одно и то же имя.

Итак, значения каждого атрибута относятся к одному из доменов, то есть атрибут может принимать значения из заданного множества. Атрибут имеет имя и значение, часто имя атрибута совпадает с именем домена.

## Вариант 21

### **Нормализация базы данных**

При проектировании реляционной базы данных необходимо решить вопрос о наиболее эффективной структуре данных. Основные цели, которые при этом преследуются: обеспечить быстрый доступ к данным в таблицах; исключить ненужное повторение данных (избыточность), которая может явиться причиной ошибок при вводе и нерационального использования дискового пространства компьютера; обеспечить целостность данных таким образом, чтобы при изменении одних объектов автоматически происходили соответствующие изменения связанных с ними объектов. Так, корректной является схема базы данных, в которой отсутствуют нежелательные зависимости между атрибутами отношений. Реляционная база данных - это набор отношений. Но не просто набор, а нормализованный набор отношений.

Классическая технология проектирования реляционных баз данных связана с *теорией нормализации*, основанной на анализе функциональных зависимостей между атрибутами отношений. Понятие функциональной зависимости является фундаментальным в теории нормализации реляционных баз данных. Функциональные зависимости определяют устойчивые отношения между объектами и их свойствами в рассматриваемой предметной области. Именно поэтому процесс поддержки функциональных зависимостей, характерных для данной предметной области, является базовым для процесса проектирования.

Нормализация – метод создания набора отношений с заданными свойствами на основе требований к данным, установленных в некоторой организации. Процесс приведения реляционных таблиц к стандартному виду называется нормализацией. В теории реляционных баз, к примеру, выделяют следующую последовательность нормальных форм: первая нормальная форма (1NF); вторая нормальная форма (2NF); третья нормальная форма (3NF); нормальная форма Бойса-Кодда (BCNF); четвертая нормальная форма (4NF); пятая нормальная форма, или форма проекции-соединения. При практическом проектировании баз данных используются первые три нормальные формы.

Процесс проектирования с использованием декомпозиции представляет собой процесс последовательной нормализации схем отношений, при этом каждая последующая итерация соответствует нормальной форме более высокого уровня и обладает лучшими свойствами по сравнению с предыдущей. Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

Таким образом, основные свойства нормальных форм: каждая следующая нормальная форма в некотором смысле улучшает свойства предыдущей; при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

## Вариант 22

### Состояние базы данных

В основе классического процесса проектирования лежит последовательность переходов от предыдущей нормальной формы к последующей. Однако в процессе декомпозиции мы сталкиваемся с проблемой обратимости, то есть возможности восстановления исходной схемы. Точнее, декомпозиция должна сохранять эквивалентность схем базы данных при замене одной схемы на другую. Схемы базы данных называются эквивалентными, если содержание исходной базы данных может быть получено путем естественного соединения отношений, входящих в результирующую схему, и при этом в исходной базе данных не появляются

новые кортежи. В частности, при выполнении эквивалентных преобразований сохраняется множество исходных фундаментальных функциональных зависимостей между атрибутами отношений.

Функциональные зависимости определяют не текущее состояние базы данных, а все возможные ее состояния, то есть они отражают те связи между атрибутами, которые присущи реальному объекту, который моделируется с помощью базы данных. Поэтому определить функциональные зависимости по текущему состоянию базы данных можно только в том случае, если экземпляр базы данных содержит абсолютно полную информацию, а именно никаких добавлений и модификации базы данных не предполагается. В реальной жизни это требование невыполнимо, поэтому набор функциональных зависимостей задает разработчик, системный аналитик, исходя из глубокого системного анализа предметной области.

Из-за небрежного проектирования базы данных появляется избыточность данных (повторение), которая приводит не только к потере лишнего места, но и может вызвать нарушение целостности, точнее привести к противоречивости данных в базе данных. Несоответствие между данными (противоречивость) называются аномалиями. Существует три вида аномалий. Например, аномалия обновления – это противоречивость данных, вызванная их избыточностью и их частичным обновлением; аномалия удаления – непреднамеренная потеря данных, вызванная удалением других данных; аномалия ввода – невозможность ввести данные в таблицу, вызванная отсутствием других данных.

Реляционная таблица находится в первой нормальной форме (1НФ), если никакое значение атрибута не является множеством значений или повторяющейся группой. Иногда говорят, что атрибуты таблицы, которая находится в 1НФ, являются скалярными величинами. Иначе говоря, отношение находится в первой нормальной форме тогда и только тогда, когда на пересечении каждого столбца и каждой строки находятся только элементарные значения атрибутов. В некотором смысле это определение избыточно, потому что собственно оно определяет само отношение в теории реляционных баз данных. Однако, в силу исторически сложившихся обстоятельств и для преемственности такое определение первой нормальной формы существует. Определение Кодда реляционной таблицы содержит условие, согласно которому реляционная таблица должна удовлетворять первой нормальной форме.

Вариант 23

### **Нормальные формы реляционной таблицы**

Функциональная зависимость – значение атрибута в кортеже однозначно определяет значение другого атрибута в кортеже. Формально функциональная зависимость определяется следующим образом: если  $A$  и  $B$  –

атрибуты в таблице R, то запись обозначает, что если два кортежа в таблице R имеют одно и то же значение атрибута A, то они имеют одно и то же значение атрибута B. Это определение также применимо, если A и B – множества столбцов, а не просто отдельные столбцы. Обозначение читается следующим образом: A функционально определяет B. Атрибут в левой части ФЗ называется *детерминантом*. *Детерминант* – атрибут (атрибуты), значение которого определяет значения других атрибутов кортежей. Реляционная таблица находится во *второй нормальной форме (2НФ)*, если никакие не ключевые атрибуты не являются функционально зависимыми лишь от части ключа.

Реляционная таблица имеет третью нормальную форму (3НФ). Любая таблица, удовлетворяющая 3НФ, также удовлетворяет и 2НФ. Если таблица не соответствует 3НФ, то возникает избыточность данных, которая также приводит к всевозможным аномалиям. Поскольку 3НФ-таблицы всегда удовлетворяют 2НФ, достаточно пользоваться критерием третьей нормальной формы. Если каждый детерминант в таблице является ключом этой таблицы, то она удовлетворяет первой, второй и третьей нормальным формам. Это упрощает процесс нормализации, так как нужно проверить только один критерий. Одним словом, данное определение третьей нормальной формы соответствует в некоторых источниках нормальной форме Бойса-Кодда. В большинстве случаев достижение третьей нормальной формы считается достаточным для реальных проектов баз данных, однако в теории нормализации существуют нормальные формы высших порядков, которые уже связаны не с функциональными зависимостями между атрибутами отношений, а отражают более тонкие вопросы семантики предметной области и связаны с другими видами зависимостей.

Таким образом, проблема многозначных зависимостей возникает в связи с многозначными атрибутами, поэтому она решается размещением каждого многозначного атрибута в свою собственную таблицу вместе с ключом, от которого атрибут зависит.

## Вариант 24

### **Концептуальное и реляционное моделирование данных**

Главное назначение базы данных состоит в обеспечении пользователя точной информацией. Поэтому важно, чтобы структура базы данных была логичной и не имела изъянов. Чем сложнее модель базы данных, тем труднее правильно ее спроектировать. Эта сложность растет с добавлением объектных множеств, конкретизаций, составных объектов и отношений. В частности, графический подход (схема концептуальной модели) сильно повышает вероятность получения точных моделей по сравнению с текстовым подходом реляционной схемы.

Концептуальные модели данных значительно легче понимать, чем реляционные модели, поскольку они лучше соответствуют нашему естественному взгляду на вещи, отраженному в языке в виде употребления глаголов и существительных (объектных множеств). К примеру, информация представляется в графическом виде, визуальное представление упрощает понимание структуры.

В реляционной же модели единственными средствами являются реляционные таблицы и внешние ключи, все объектные множества и отношения моделируются одинаковым образом. С одного взгляда на реляционную схему не просто понять, какие таблицы представляют объекты, а какие – отношения. Анализируя реляционную модель, вы мысленно рисуете картинку, которая связывает таблицы. Иными словами, для понимания реляционной модели мысленно проделывается та же работа, которая выполняется при концептуальном моделировании. Поэтому создание концептуальной модели и последующее преобразование ее в нормализованную реляционную модель стоит затраченных усилий, иначе в сложных ситуациях моделирования с большей вероятностью можно допускать серьезные ошибки. Конечно, для простой базы данных создать реляционную модель, возможно, не сложнее, чем создавать концептуальную модель.

Теория нормализации, которая была рассмотрена ранее применительно к реляционной модели, применима и к модели «объект-связь». Алгоритм приведения семантической модели к 5-й нормальной форме может быть следующим:

- проанализировать схему на присутствие объектов, которые моделируют несколько взаимосвязанных объектов реального мира (именно это соответствует ненормализованным отношениям) и разделить каждое из этих объектов на несколько новых, установить между ними соответствующие связи, полученная схема будет находиться в первой нормальной форме;

- проанализировать все сущности, имеющие составные первичные ключи, на наличие неполных функциональных зависимостей непервичных атрибутов от атрибутов возможного ключа. Если такие зависимости обнаружены, то разделить данные сущности на два, определить для каждой сущности первичные ключи и установить между ними соответствующие связи. Полученная схема будет находиться во второй нормальной форме;

- проанализировать все сущности на наличие детерминантов, которые не являются возможными ключами. При обнаружении подобных расщепить сущность на две, установив между ними соответствующие связи. Полученная схема соответствует третьей нормальной форме;

- проанализировать все сущности на наличие многозначных зависимостей. Если обнаружатся сущности, у которых имеется более одной многозначной зависимости, то расщепить такие сущности на две, установив между ними соответствующие связи.

Полученная таким образом схема будет находиться в четвертой нормальной форме.



## Двухуровневые модели

Двухуровневая модель фактически является результатом распределения пяти функций между двумя процессами, которые выполняются на двух платформах: на клиенте и на сервере. В чистом виде почти никакая модель не существует. К примеру, наиболее характерной особенностью каждой двухуровневой модели является Модель удаленного управления данными, точнее *модель файлового сервера* (File Server, FS). В этой модели СУБД, а также функции управления всеми информационными ресурсами находится на клиенте. На сервере располагаются файлы с данными, и поддерживается доступ к файлам. Запрос клиента формулируется в командах DML. СУБД переводит этот запрос в последовательность файловых команд. Таким образом, каждая файловая команда вызывает перекачку блока информации на клиента.

В *модели удаленного доступа* к данным (Remote Data Access, RDA) база данных хранится на сервере. На сервере же находится ядро СУБД. На клиенте располагается презентационная логика и бизнес-логика приложения. Точнее, клиент обращается к серверу с запросами на языке SQL.

*Модель сервера баз данных* поддерживают большинство современных СУБД. Основу данной модели составляет механизм хранимых процедур как средство программирования SQL-сервера, механизм триггеров как механизм отслеживания текущего состояния информационного хранилища и механизм ограничений на пользовательские типы данных, который иногда называется механизмом поддержки доменной структуры. В этой модели бизнес-логика разделена между клиентом и сервером. На сервере бизнес-логика реализована в виде хранимых процедур - специальных программных модулей, которые хранятся в БД и управляются непосредственно СУБД. Клиентское приложение обращается к серверу с командой запуска хранимой процедуры, а сервер выполняет эту процедуру и регистрирует все изменения в БД, которые в ней предусмотрены. Сервер возвращает клиенту данные, релевантные его запросу, которые требуются клиенту либо для вывода на экран, либо для выполнения части бизнес-логики, которая расположена на клиенте. Это означает, что трафик обмена информацией между клиентом и сервером резко уменьшается.

Централизованный контроль в модели сервера баз данных выполняется с использованием механизма триггеров. Механизм использования триггеров предполагает, что при срабатывании одного триггера могут возникнуть события, которые вызовут срабатывание других триггеров. Этот мощный

инструмент требует тонкого и согласованного применения, чтобы не получился бесконечный цикл срабатывания триггеров. В данной модели сервер является активным, потому что не только клиент, но и сам сервер, используя механизм триггеров, может быть инициатором обработки данных.

### **Семестровая работа студента № 3**

**Тема:** виды и способы развития информации в тексте.

**Цель:** применить на практике знания о видах информации в научном тексте.

**Задачи:**

1) В каждом микротексте указать основную и виды дополнительной информации.

2) Составить толковый терминологический словарь к анализируемому тексту.

3) Пересказать текст (распределение вариантов остаётся согласно выполненной ранее СРС №2).

### **Рекомендуемая литература для выполнения СРС № 3**

1 Русский язык: учебное пособие для студентов казахских отделений университета (бакалавриат)/ Под ред. К.К.Ахмедьярова, Ш.К.Жаркынбековой. – Алматы: Қазақ университеті, 2009. – 226 с.

2. Мейрбекова М.М., Е.Е. Пособие по русскому языку для студентов механико-математического факультета университета. – Алматы: Қазақ университеті, 2013. – 178 с.

3 Мухамадиев Х.С. Пособие по научному стилю речи: для казахских отделений университета. - 3-е изд. – Алматы: Қазақ университеті, 2011.- 210 с.

4 Ишлинский А.Ю. Политехнический словарь. – М.: Советская энциклопедия, 1980. – 656 с.

5 Даль В.И. Большой иллюстрированный толковый словарь русского языка: современное написание. Около 1500 илл. – М.: «АСТ – Астрель - Хранитель», 2008. – 352 с.

### **Семестровая работа № 4**

**Тема:** аннотирование научного текста.

**Цель:** закрепить навыки аннотирования.

**Задачи:**

1) Подобрать текст из учебно-научной литературы по своей специальности объемом 7-8 страниц (см. рекомендательный список).

2) Составить толковый словарь узкоспециальных терминов, содержащихся в выбранном тексте.

3) Написать аннотацию текста (приложить оттиск выбранного текста).

## Рекомендательный список вариантов для выполнения СРС № 4

I По книге: Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. – Изд. 4-е, перераб. и доп. – СПб: Издательство «Профессия», 2004. - 752 с.

- 1 Виды систем автоматического управления (с. 9-18).
- 2 Примеры дискретных и релейных автоматических систем (с. 22-29).
- 3 Программы и алгоритмы управления (с. 29-35).
- 4 Линеаризация дифференциальных уравнений систем автоматического управления (с. 40-48).
- 5 Общие сведения об устойчивости (с.115-123).
- 6 Устойчивость систем с распределенными параметрами (с. 154-162).
- 7 Использование преобразований Фурье, Лапласа и Карсона-Хевисайда (с.173-181).
- 8 Использование вычислительных машин (с.181-190).
- 9 Общие методы повышения точности систем автоматического управления (с.238-247).
- 10 Теория инвариантности и комбинированное управление (с.247-255).
- 11 Улучшение качества процесса управления. Обратные связи (с.269-279).
- 12 Методы повышения запаса устойчивости (с.279-286).
- 13 Нахождение функции веса и построение переходных процессов (с.387-395).
- 14 Импульсные системы. Уравнение состояния (с.427-430).
- 15 Приближенное исследование переходных колебательных процессов (с.631-639).
- 16 Симметричные одночастотные вынужденные колебания (с. 646-653).
- 17 Несимметричные вынужденные колебания с медленно меняющейся составляющей (с.653-660).
- 18 Системы с широтно-импульсной модуляцией (с.695-703).
- 19 Адаптивные системы. Системы экстремального управления (с.723-733).
- 20 Самонастраивающиеся системы (с.733-741).

II Статьи в периодических изданиях.

1 Колбанев М.О., Татарникова Т.М. Информационный объем базовых информационных процессов// Информационно-управляющие системы. – 2014. - № 4(71). – С.42-48.

2 Дьячук П.П., Логинов Д.А., Карабалыков С.А. Синергетический подход к управлению учебной деятельностью в вербальных проблемных средах//Информационно-управляющие системы. – 2014. - № 3(70). – С.118-125.

3 Беззатеев С.В., Волошина Н.В., Санкин П.С. Методика расчета надежности слабых систем, учитывающая угрозы информационной

безопасности//Информационно-управляющие системы. – 2014. - № 3(70). – С.78-84.

4 Чубраева Л.И., Шишлаков А.В. Синтез электромеханических систем автоматического управления при аналитической аппроксимации характеристик нелинейных элементов//Информационно-управляющие системы. – 2014. - № 2(69). – С. 2-9.

5 Браншитов С.А., Тумченко Д.А., Ширванял А.М. Вопросы автоматизации управления поездной работой//Информационно-управляющие системы. – 2014. - № 1(68). – С.32-43.

6 Соколов А. Совместное использование Autodesk Inventor и Autodesk Vault Professional в машиностроении// САПР и графика. – 2015. - № 9(227). – С.6-14.

### **Семестровая работа студента № 5**

**Тема:** реферирование.

**Цель:** презентация обзорного информативного реферата.

**Задачи:**

- 1) Определить тему обзорного реферата из предложенного списка.
- 2) Подобрать необходимую литературу для раскрытия темы.
- 3) Написать реферат на основе сопоставления, сравнения и обобщения разных источников.
- 4) Сделать презентацию своей работы.

Перечень тем рефератов.

1. Профессия инженера вчера, сегодня, завтра.
2. Профессионал и специалист: кто есть кто?
3. Крупнейшее изобретение инженерной мысли, повлиявшее на развитие цивилизации – это...
4. Кто изобрел колесо?
5. Человек и машина: история взаимоотношений.
6. Прорывные технологии в автоматизации и управлении производственными процессами энергоснабжения.
7. Прорывные технологии в автоматизации и управлении производственными процессами теплоснабжения.
8. Системы автоматизированного проектирования (САПР). Их место в оптимизации современного производства.
9. Системы автоматизированного управления (САУ) как основа перехода общества технологиям будущего.
10. Комплексная автоматизация производственных процессов: история, современное состояние и перспективы.
11. Использование ЭВМ для решения проблем автоматического управления.

12. Оптимизация сферы делопроизводства контроля за документооборотом современного предприятия.
13. Оптимальные параметры применения компьютерных программ в управлении производством.
14. Вопросы автоматизации и управления в экономической сфере.
15. Проблемы «кусочной» автоматизации задач управления.
16. Традиционные подходы и принципы системы автоматизированного управления.
17. Интеллектуальная составляющая систем автоматического управления.
18. Автоматизация и управление современным предприятием: тенденции и проблемы.
19. Применение объектно-ориентированного анализа при разработке сложных технических систем.

### **Рекомендуемая литература для выполнения СРС № 5**

- 1 Русский язык: учебное пособие для студентов казахских отделений университета (бакалавриат)/ Под ред. К.К.Ахмедьярова, Ш.К.Жаркынбековой. – Алматы: Қазақ университеті, 2009. – 226 с.
- 2 Саньярова Н.С. Русский язык. Практикум по культуре речи: Учебное пособие. – Алматы: НАО «АУЭС», 2013. – 480 с.
- 3 Мейрбекова М.М., Хайрушева Е.Е. Пособие по русскому языку для студентов механико-математического факультета университета. – Алматы: Қазақ университеті, 2013. – 178 с.
- 4 Нурмаханова М.К. Русский язык для бакалавриата всех специальностей: Учебное пособие. – Алматы: АУЭС, 2014. – 95 с.
- 5 Смирнова Ю.Г. Грамматика технического текста: Учебное пособие. – Алматы: АУЭС, 2013. - 200 с.
- 6 Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. – Изд. 4-е, перераб. и доп. – СПб: Издательство «Профессия», 2004. - 752 с.
- 7 Беляев Г.Б., Кузицин В.Ф., Смирнов Н.И. Технические средства автоматизации в теплоэнергетике. – М.: Энергоиздат, 1982. – 320 с.

Список периодических изданий, рекомендуемых для отбора статьи:

- 1) Техника молодежи//Санкт-Петербург: Изд.ООО «Девиз», 2015.
- 2) Наука и жизнь//М.: «Наука и жизнь», 2015.
- 3) Инженер//М.: ГУП МО КТ «Раменская типография», 2015.
- 4) Знание - сила//М.: АО «Первая образцовая типография», 2015.
- 5) САПР и графика//М.: ООО Компьютер Пресс, 2015.

### **Семестровая работа № 6**

**Тема:** содержание этики речевого поведения в деловой коммуникации.

**Цель:** демонстрация владения образцами речевого этикета в деловой сфере.

Задачи:

1. Выбрать тему из предложенных вариантов.
2. Составить словарь ситуативных речевых образцов (не менее 20 единиц).
3. Предъявить знание формул этикета в устном диалоге с преподавателем.

Темы диалогов:

- 1) Собеседование при приеме на работу по специальности после окончания вуза.
- 2) Разговор с директором фирмы, давшей объявление о вакансиях инженеров.
- 3) Диалог с ректором университета о переводе на другую специальность.
- 4) Разговор с начальником о повышении оклада.
- 5) Обращение старосты к преподавателю о продлении срока сдачи СРС.
- 6) Интервью декана факультета студенту для газеты «Энтел».
- 7) Беседа с организатором курсов по вождению автомобиля, на которых Вы хотели бы заниматься.

### **Рекомендуемая литература для выполнения СРС № 6**

1 Русский язык и культура речи [Текст]: Учебник/Под ред. проф. О.Я.Гойхмана. – 2- изд., перераб. и доп. - М.: ИНФРА-М, 2009. – 240 с.

2 Салагаев В.Г. Культура делового общения. Деловая риторика. Деловые документы [Текст]: Учебное пособие/ В.Г.Салагаев. – Алматы: Раритет, 2000. – 200 с.

3 Салагаев В.Г. Риторика. Технология сотрудничества: Учебное пособие.– Астана: Дарын, 2007. – 304 с.

4 Саньярова Н.С. Русский язык. Практикум по культуре речи: Учебное пособие. – Алматы: НАО «АУЭС», 2013. – 480 с.

5 Нурмаханова М.К. Русский язык для бакалавриата всех специальностей: Учебное пособие. – Алматы: АУЭС, 2014. – 95 с.

6 Мейрбекова М.М., Хайрушева Е.Е. Пособие по русскому языку для студентов механико-математического факультета университета. – Алматы: Қазақ университеті, 2013. – 178 с.

## Список литературы

- 1 Беляев Г.Б., Кузищин В.Ф., Смирнов Н.И. Технические средства автоматизации в теплоэнергетике. – М.: Энергоиздат, 1982. – 320 с.
- 2 Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. – Изд. 4-е, перераб. и доп. – СПб: Издательство «Профессия», 2004. – 752 с.
- 3 Даль В.И. Большой иллюстрированный толковый словарь русского языка: современное написание. Около 1500 илл. – М.: «АСТ – Астрель - Хранитель», 2008. – 352 с.
- 4 Ишлинский А.Ю. Политехнический словарь. – М.: Советская энциклопедия, 1980. – 656 с.
- 5 Мейрбекова М.М., Хайрушева Е.Е. Пособие по русскому языку для студентов механико-математического факультета университета. – Алматы: Қазақ университеті, 2013. – 178 с.
- 6 Мухамадиев Х.С. Пособие по научному стилю речи: для казахских отделений университета. - 3-е изд. – Алматы: Қазақ университеті, 2011.- 210 с.
- 7 Нурмаханова М.К. Русский язык для бакалавриата всех специальностей: Учебное пособие. – Алматы: АУЭС, 2014. – 95 с.
- 8 Пособие по научному стилю речи. Для вузов технического профиля/ Под ред. И.Г.Проскуряковой. – 2-е изд., доп. и перераб. – М.: Флинта, 2004. – 220 с.
- 9 Русский язык: Учебное пособие для студентов казахских отделений университетов (бакалавриат). - Под редакцией К.К.Ахмедьярова, Ш.К.Жаркынбековой. – Алматы: Қазақ университеті, 2009. – 226 с.
- 10 Русский язык: экспериментальное пособие для нефилологических специальностей вузов государств-участников СНГ. – М.: АНО «Институт национальных проблем образования», 2007. – 336 с.
- 11 Русский язык и культура речи [Текст]: Учебник/Под ред. проф.О.Я.Гойхмана. – 2- изд., перераб. и доп.- М.: ИНФРА-М, 2009. – 240 с.
- 12 Салагаев В.Г. Культура делового общения. Деловая риторика. Деловые документы [Текст]: Учебное пособие/ В.Г.Салагаев. – Алматы: Раритет, 2000. – 200 с.
- 13 Салагаев В.Г. Риторика. Технология сотрудничества: Учебное пособие.– Астана: Дарын, 2007. – 304 с.
- 14 Саньярова Н.С. Русский язык. Практикум по культуре речи: Учебное пособие. – Алматы: НАО «АУЭС», 2013. – 480 с.
- 15 Смирнова Ю.Г. Грамматика технического текста: Учебное пособие. – Алматы, 2013. – 200 с.

- 16 Федосюк, М.Ю. и др. Русский язык для студентов-нефилологов [Текст]: Учебное пособие / М.Ю. Федосюк. – 4-е изд. – М.: Флинта: Наука, 2000. – 256 с.
- 17 Техника молодежи//Санкт-Петербург: Изд.ООО «Девиз», 2015.
- 18 Наука и жизнь//М.: «Наука и жизнь», 2015.
- 19 Инженер//М.: ГУП МО КТ «Раменская типография», 2015.
- 20 Знание - сила//М.: АО «Первая образцовая типография», 2015.
- 21 САПР и графика//М.: ООО Компьютер Пресс, 2015.



## Содержание

Введение.....	3
Семестровая работа студента № 1.....	4
Семестровая работа студента № 2.....	4
Варианты текстов для выполнения СРС № 2 и СРС № 3.....	5
Семестровая работа студента № 3.....	34
Семестровая работа студента № 4.....	34
Семестровая работа студента № 5.....	36
Семестровая работа студента № 6.....	37
Список литературы.....	39

Досмаханова Райкул Амандыковна

РУССКИЙ ЯЗЫК

Методические указания и варианты по выполнению семестровых работ  
для студентов специальности 5В070200 – Автоматизация и управление

Редактор

Специалист по стандартизации Н. К. Молдабекова

Подписано в печать \_ \_ \_

Тираж 50 экз.

Объем 2,6 уч.-изд. л.

Формат 60x84 1/16

Бумага типографская №1

Заказ \_\_\_\_\_ Цена 1250 тг

Копировально-множительное бюро  
некоммерческого акционерного общества  
«Алматинский университет энергетики и связи»  
050013 Алматы, Байтурсынова,126