



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Факультет	<u>Энергетика и нефтегазопромышленность</u>
Кафедра	<u>АММ НГК</u>
Направление	<u>15.03.04 Автоматизация технологических процессов и производств (бакалавриат)</u>

КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ

«Алгоритмизация производств нефтегазового комплекса»

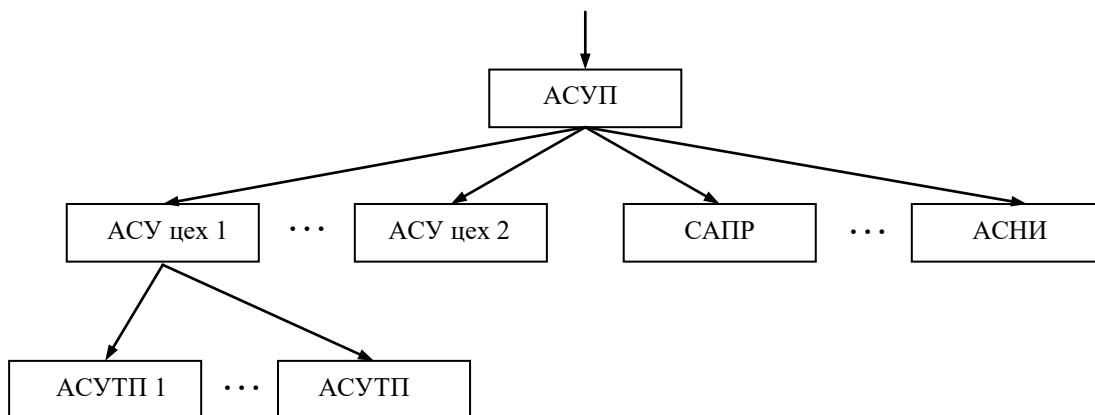
Ростов-на-Дону

2017

1 Автоматизированные информационно – управляющие системы (АИУС)

1.1 Информационный банк данных. Автоматизированные банки данных

В настоящее время получили распространение различного рода автоматические системы. Каждая из таких систем реализует специфический набор задач и функций, которые как раз и определяют структуру системы, аппаратные средства и программные средства этой системы.



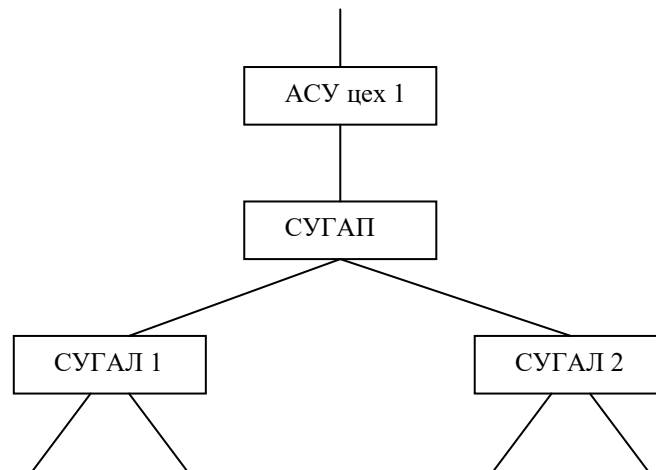
ОАСУ — отраслевая автоматизированная система управления.

АСУП — автоматизированная система управления предприятия.

АСУТП — автоматизированная система управления технического процесса.

СУГАЛ — система управления гибкой автоматизированной линии.

СУГАП — система управления гибкого автоматизированного производства.



В рамках каждого типа автоматизированных систем обычно можно выделить целый ряд так называемых функциональных подсистем, которые реализуют конкретные задачи по управлению объектом.

Например, в рамках АСУП можно выделить подсистемы — ОКП, БУ, УК, МТС, СиРП, УиОП, ...

ОКП — оперативно календарного планирования

БУ — бухгалтерского учета

УК — учета кадров

МТС — материально технического снабжения

СиРП — сбыта и реализации продукции

УиОП — управления и оперативного планирования

В рамках АСУТП выделяются подсистемы — СиПОИ, РиСП, ОР, ...

СиПОИ — сбора и первичной обработки информации

РиСП — регулирования и стабилизации параметров

ОР — организации режимов

Кроме функциональных подсистем имеют место обеспечивающие подсистемы. С их помощью обеспечивается нормальное функционирование системы. Могут включать — ПО, ИО, ТО, ...

ПО — программное обеспечение

ИО — информационное обеспечение

ТО — техническое обеспечение

Любая информационная система содержит ИО.

Каждая из функциональных подсистем использует ту или иную информацию об объекте (предметной области) в ранее создаваемых автоматизированных системах эта информация хранилась обычно в виде массивов данных, которые и использовались программами той или иной функциональной подсистемой.

Более перспективным в настоящее время считается подход когда в рамках автоматизированной системы выделяется специальная обеспечивающая подсистема (ИО), в которой концентрируется вся информация и из которой функциональные подсистемы могут получать данные необходимые для ее функционирования. Ядром такой обеспечивающей подсистемы (ИО) являются автоматизированные банки данных (АБД).

Рассмотрим примеры АСУП и АСУТП.

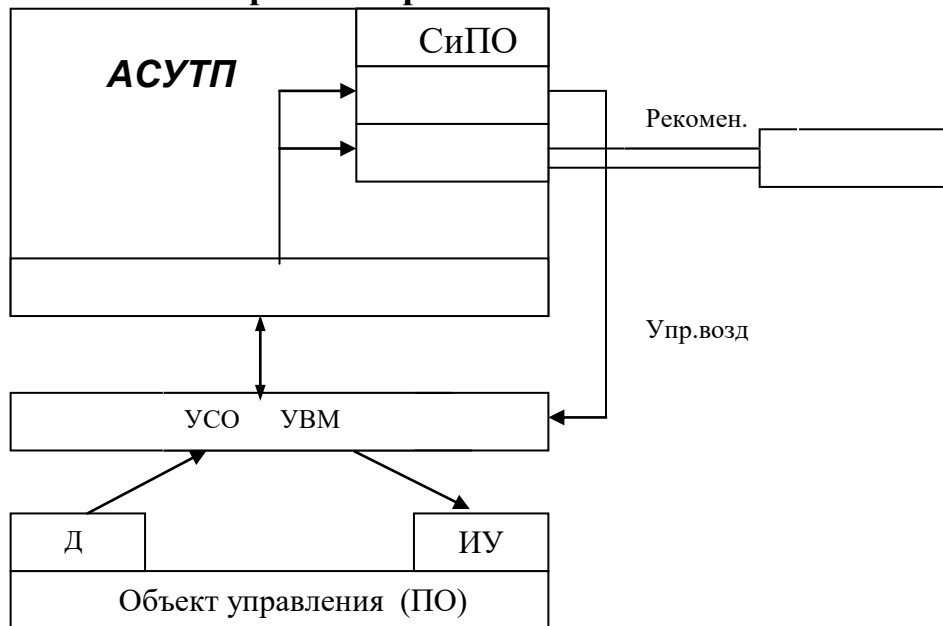
Пример 1. Рассмотрим АСУП предприятия с выделением подсистемы. Для такой системы имеют место следующие особенности:

1. Преобладают задачи или функции связанные с большим количеством расчетов.
2. Структура хранимой информации не отличается большой сложностью, но объемы ее велики.
3. Имеет место жестко заданный перечень запросов, которые определяются функциями автоматизированной системы.
4. Информационная подсистема должна формировать документы и отчеты с заданными формами.

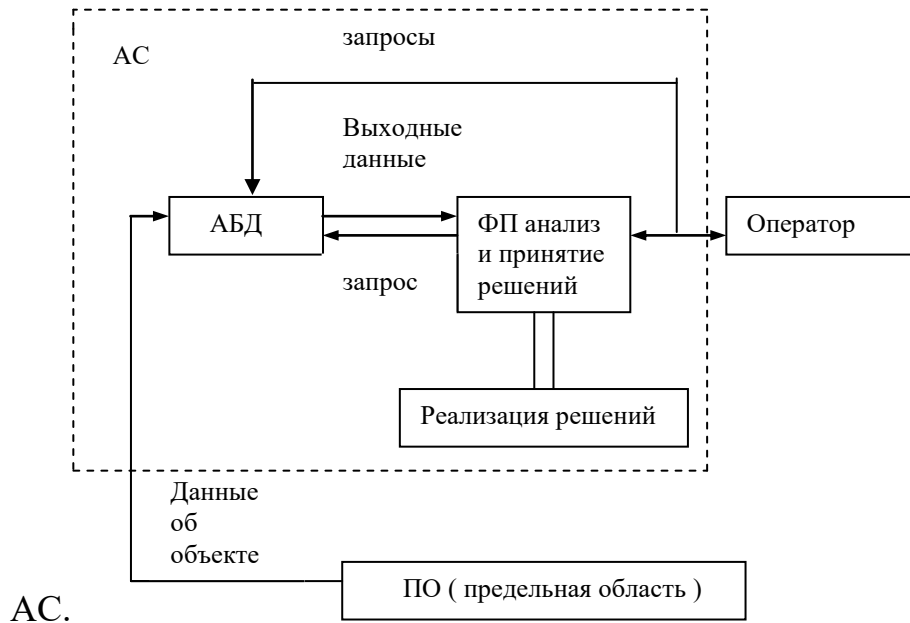
АСУП



Система принятия решения



Общая структура



Спецификой такого рода системы АСУТП является следующее:

1. Должен быть обеспечен высоко оперативный режим работы в реальном масштабе времени.
2. Наличие сложных расчетов для реализации моделирования поведения объекта и выработке правильных управляющих воздействий.
3. Большой объем вычислений.

В общем случае АБД — это подсистема АС включающая в свой состав комплекс специальных методов и средств (математических, технических, программных, лингвистических и организационных) для поддержки динамической информационной модели предметной области с целью решения задач управления или обслуживания потребителей.

Обычно к АБД предъявляются требования:

1. Система должна удовлетворять информационной потребности пользователя.
2. Система должна обеспечивать заданный уровень достоверности хранящейся информации.
3. Система должна обеспечивать требуемую производительность при обработке запроса.

4. Система должна обеспечивать доступ к данным пользователей в соответствии с их полномочиями.

5. Система должна обеспечивать удобство обращения пользователя за информацией.

6. Система должна обеспечивать возможность реорганизации и расширения.

1.2 Информационные системы. Основные понятия.

1.2.1 Информация и данные.

В окружающем нас мире имеют место либо физические тела, либо физические поля. Последние находятся в состоянии непрерывного изменения, которые сопровождаются обменом энергией и выделением сигналов в окружающей среде. При взаимодействии этих сигналов с другими телами или объектами, в последних также возникают определенные изменения их свойств или состояний. Это и называется регистрацией сигнала.

Эти изменения можно наблюдать или измерять. В результате этого процесса образуются данные. Данные – это зарегистрированные сигналы. Данные несут в себе информацию о событиях реального мира, однако они не тождественны информации.

Предположим, что нужно оценить показания бегуна на заданной дистанции. Если есть секундомер, то можно зафиксировать начальное и конечное время. В итоге мы замеряем величину перемещения стрелки за время пробега. Это и есть регистрация данных. Однако для того чтобы данные о перемещении стрелки дали информацию, необходимо наличие метода пересчета одной физической величины в другую. Во первых, для этого нужно знать цену деления шкалы секундомера, а во вторых нужно знать, как умножается цена деления на число деления (методы умножения).

Предположим, у нас есть радиоприемник. Прослушивая передачу на незнакомом языке мы регистрируем звуковые данные, но не получаем информацию так как не владем методом преобразования данных в известные нам понятия. Если же мы запишем передачу на магнитофон и попросим переводчика, то он владея методом перевода может извлечь информацию и изложить результат на понятном нам языке.

Информация – это продукт взаимодействия данных и адекватным им методам.

Определение не является единственным. В настоящее время нет строгого общепризнанного определения информации, хотя имеются различные понятия об информации. Причем оказывается, что в разных научных дисциплинах это понятие интерпретируется по-разному. В результате может оказаться, что понятие введенное в рамках одной научной дисциплины плохо подходит для другой. В частности в литературе по естественным наукам имеет место представление об информации как о совокупности данных повышающих уровень знаний об окружающем мире. Такое определение для дисциплин, как информационно техническое, информационно научное, не подходит, так как основывается на таком понятии как знание. В тоже время средства вычислительной техники способны без участия человека обрабатывать информацию о каком либо знании или незнании речи не идет.

1.2.2.Свойства информации

1) Динамический характер информации, т.е. информация не является статическим объектом. Она изменяется во времени и существует только в момент взаимодействия данных и метода.

2) Объективность и субъективность информации. Данные сами по себе являются объективными, т.к. они являются результатом регистрации объективно существующих сигналов. В тоже время методы являются

субъективными. Но тот метод, который используется для обработки с данными, может быть субъективным, если другой работает с теми же данными. Более объективной принято считать информацию, в которую методы вносят меньше субъективных элементов.

Так, например результат наблюдения фотоснимка какого либо объекта дает более объективную информацию, чем результат наблюдения рисунка этого же объекта.

3) Полнота информации. Полнота информации во многом характеризует качество информации и определяет достаточность данных для принятия решения.

4) Достоверность информации. Данные возникают в момент регистрации сигнала, но не все сигналы являются полезными, т.е. всегда имеет место какой-то уровень посторонних сигналов. В результате полезные данные сопровождаются определенным уровнем информационных шумов. При увеличении уровня шумов достоверность информации снижается. И в таком случае для получения того же количества информации и того же сигнала необходимо использовать либо больше данных, либо более совершенные методы.

5) Адекватность информации — это степень соответствия реальному процессу или объекту.

6) Доступность информации — это мера возможности получить ту или иную информацию. На степень доступности влияют одновременно и степень доступности к данным и доступность к адекватным методам для их интерпретации.

7) Актуальность информации — степень соответствия информации текущему моменту. Поскольку информационные процессы происходят во времени, то адекватная и достоверная, но устаревшая информация может приводить к ошибочным решениям.

1.2.3 Носители данных и обработка данных

В настоящее время используются в качестве носителей данных: бумага, магнитные диски, компакт диски (оптические), полупроводниковая память, флешь диски и т.д. Их можно охарактеризовать по следующим критериям:

1. Разрешающая способность(плотность записи данных на носителе)
2. Динамический диапазон: $\ln(\max/\min)$ регистрации сигнала

Операции с данными: преобразования из одного вида в другой с помощью разных методов обработки данных, при этом может использоваться:

- 1) Сбор данных, т.е. накопление информации с целью обеспечения достаточной ее полноты для принятия решения;
- 2) Формализация данных, это приведение данных, поступающих от разных источников к одной форме, чтобы сделать их сопоставимыми между собой;
- 3) Фильтрация данных, это отсечение ненужных данных, в которых нет необходимости при принятии решения, при этом должна повышаться достоверность и адекватность данных, а уровень шума снижаться;
- 4) Сортировка, упорядочение данных по заданному признаку, с целью повышения удобства использования, и повышения доступности информации;
- 5) Архивация, организация хранения данных в удобной компактной и легко доступной форме, при этом должно обеспечиваться снижение затрат на хранение;
- 6) Преобразование данных, перевод данных из одной формы в другую, или из одной структуры в другую;
- 7) Транспортировка данных, прием и передача данных между участниками информационного процесса

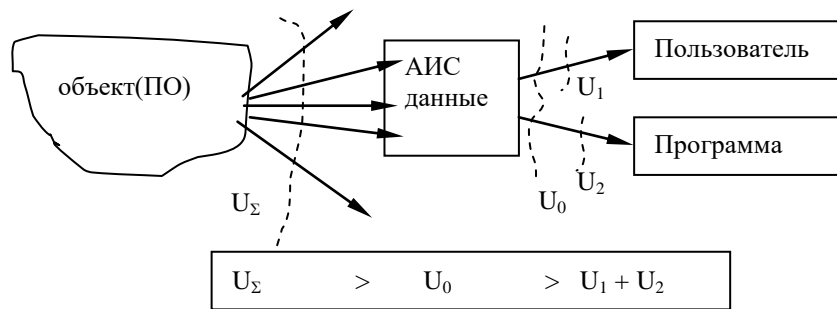
1) То, что мы имеем в виде данных об объекте или о предметной области, в ЭВМ, может, при использовании определенных методов стать информацией, т.е. приобрести смысловое содержание (семантику). Если фиксация сигнала об объекте осуществляется естественного языка, то данные и их семантика фиксируются совместно, но если используется ЭВМ для отображения состояния такого объекта, то в нем обычно данные и их интерпретация разделены, например расписание самолетов, где шапка таблицы есть семантика (интерпретация), а заполнение формы — данные:

№ рейса	Время вылета	Пункт отправления	Пункт назначения
128	12:47	Москва	Анапа
324	10:15	Москва	Новороссийск
763	17:38	Анапа	Москва
841	18:10	Новороссийск	Москва

Это связано с двумя причинами:

1. ЭВМ не имели и не имеют достаточных возможностей для обработки текста на привычном языке;
2. Память ЭВМ сравнительно дорогая, и поэтому семантика возлагается на пользователя, а хранятся только данные. Причем пользователь может не сам интерпретировать данные, а разработать для этого программу.

2) Пример: пусть имеется некий объект, или некая предметная область, информация о состоянии которого представляет интерес и есть информационная система, способная воспринимать (регистрировать) сигналы о состоянии объекта и фиксировать их в своей памяти. В таком случае данные можно определить, как информацию об объекте (ПО), зафиксированную в определенной форме, пригодной для дальнейшей обработки, передачи и хранения.



При создании АБД имеют место инфологический и даталогический аспекты проектирования.

Инфологический — употребляется при рассмотрении вопросов связанных со смысловым содержанием данных независимо от их способа представления в памяти систем; и при создании АИС выделяется этап инфологического проектирования, в его рамках решаются следующие основные вопросы:

- о каких объектах или явлениях реального мира требуется накапливать и обрабатывать информацию в АИС;
- какие основные характеристики объектов и какие взаимосвязи должны быть учтены в информационной системе;
- уточняется вводимые в ИС понятия об объектах, их характеристик и взаимосвязи.

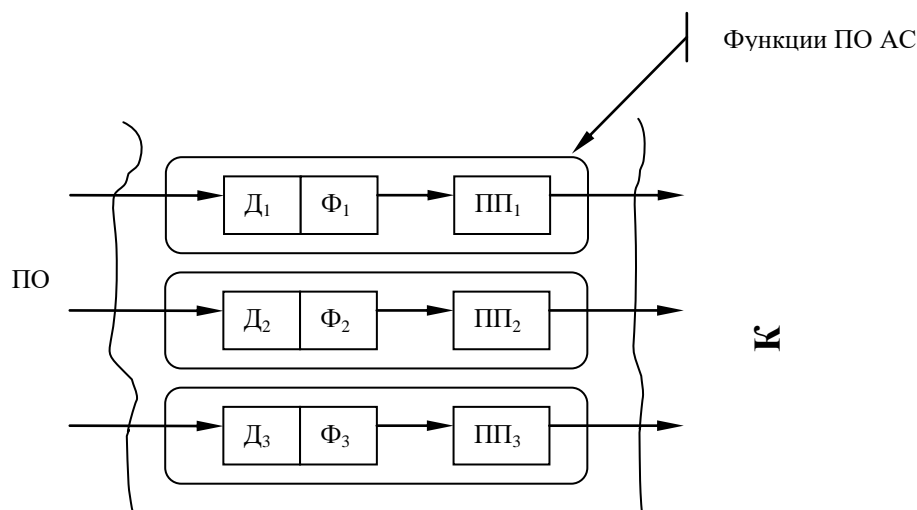
При даталогическом проектировании исходят из возможностей имеющихся средств восприятия и хранения, а также обработки данных для решения следующих вопросов:

- разработки соответствующих форм представления данных;
- при выборе моделей и метода представления и преобразования данных;
- при формулировании правил смысловой интерпретации полученных данных.

2.1 Понятия БД, системы управления БД (СУБД)

На первых периодах разработки и внедрения АИС, в их программном обеспечении использовался так называемый позадачный подход. При таком подходе, при создании сложных АС возникают проблемы:

- избыточность данных;
- сложность внесения изменений в исходные данные, так как при корректировке данных для одной программы, необходимо корректировать и для других программ;
- так как имеет место взаимосвязь между структурой данных и текстом программ, то изменение в организации или структуре данных приводит к необходимости изменения и самих прикладных программ;
- любая АС является развивающейся, а при этом внедряются новые программы, в которых если даже используется ранее применяющиеся данные, их все равно необходимо включить в новую версию программы.



Кроме них есть и другие недостатки, которые стимулировали появление новых подходов к организации данных:

1. Интеграция и централизация данных, когда все данные накапливаются, обрабатываются и хранятся централизованно, создавая динамически обновляющуюся информационную модель предметной области.
2. Обеспечение независимости прикладных программ и самих данных.

Реализация этих принципов в том или ином виде приводит к созданию единого для всех задач (прикладных) блока (базы) данных, а во-вторых – к созданию пакета управляющих программ для манипулирования этими данными.

Таким образом из состава прикладных программ $ПП_1$ и $ПП_N$ АИС, используется позадачный подход в составе АИС, построенная на основе банка данных. Во-первых в БД включены сами данные $D_1 \dots D_N$. Фрагменты ПП, обеспечивающие манипулирование данными как бы включены в состав СУБД и, наконец, сами прикладные программы, основанные на новом принципе. Из них остаются только те, которые поддерживают алгоритм конкретной задачи.

2.1.1 Преимущества АИС, на основе АБД.

По сравнению с традиционными позадачными подходами, когда каждое приложение обеспечено монопольными файлами. Централизованное управление имеет следующие преимущества:

- 1) Сокращение избыточности хранимых данных, то есть в этом случае может быть обеспечена минимально необходимая избыточность хранения данных. Если же выявляется факт использования несколькими ПП одних данных, то такие данные заносятся и хранятся в одном экземпляре (кроме резерва).
- 2) Устранение противоречивости хранимых данных. Следствием устранения избыточности данных является устранение возможности устранения противоречивости одних и тех же данных в различных приложениях.
- 3) Многоаспектное использование данных, то есть централизованное управление в полной мере позволяет реализовать принцип однократного ввода данных и многократного использования данных.

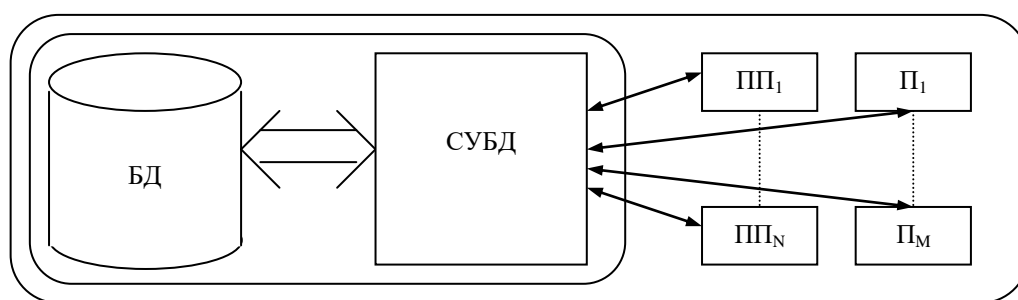
- 4) Возможность соблюдения стандартов. В настоящее время при проектировании АИС учитываются корпоративные, ведомственные, промышленные и интернациональные стандарты.
- 5) Обеспечение возможности санкционированного доступа к данным. Интеграция данных приводит к тому, что данные, использованные разными потребителями, хранятся на общих носителях и могут пересекаться самым различным образом. Этим самым повышается возможность доступа к секретным данным и т.п., а также возможность непреднамеренного искажения данных. Централизованное управление дает возможность для: - различных пользователей; - разных типов доступов (чтение, запись); - разных наборов данных установить различные механизмы защиты.
- 6) Возможность обеспечения целостности данных. Задача обеспечения данных состоит в обеспечении их правильности, точности, достоверности при хранении в БД. Например, если БД АСУ университета студент ДГТУ имеет по физике оценку '7', то система может выявить такое противоречие и информировать пользователя о наличии ошибки.

2.1.2 Независимость данных

Прежде, чем рассматривать необходимость обеспечения независимости данных, введем термины:

Хранимое поле – наименьшая единица хранимых в памяти компьютера данных.

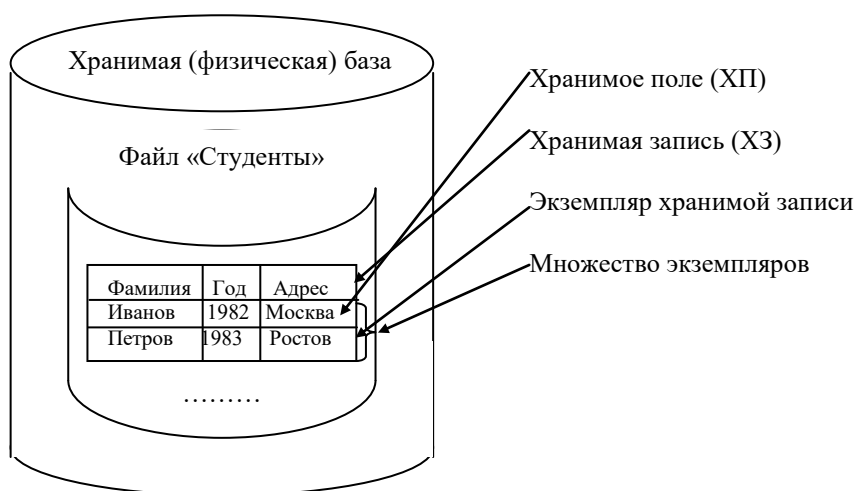
Хранимая запись – набор связанных хранимых полей.



Инфологический этап проектирования.

БД «Студенты»

На этапе инфологического проектирования для описания предметной области наиболее часто используется такое наименование, как таблица.



В системах, отличных от АБД, обычно логическая запись совпадает с хранимой записью. В связи с этим, всякие корректировки хранимых данных, необходимость в которых может возникать при эксплуатации, не приводят к необходимости корректировки логических записей, что является весьма нежелательным.

Рассмотрим аспекты хранения ФБД, которые могут быть изменены в результате тех или иных обстоятельств.

1.) Представление числовых данных. Числовое поле может храниться в различных форматах: символьном, двоичных и т.п.. При этом разработчик или администратор ФБД должен в каждом конкретном случае определить:

- подходящее основание системы исчисления;
- форму представления числа (плавающее, или с фиксированной точкой);
- тип числового поля;
- точность и количество значащих разрядов.

Каждый из этих параметров для тех или иных полей (записей) может быть изменен в процессе эксплуатации АИС. Такие изменения могут быть связаны:

- с введением новых стандартов;
- с целью повышения точности обработки данных;
- с целью повышения производительности обработки данных.

2.) Представление символьных данных. Поле в форме символьной строки может представляться в компьютере с помощью разных систем кодирования символов (ASCII, ISO, KOI-7, KOI-8). Принятая система кодирования также может быть изменена.

3.) Единицы измерения для числовых данных. Например, вместо дюймов нужны сантиметры, вместо мм р.с. – паскали. Особенно актуален вопрос о выборе единиц измерения для АИС в составе АСУТП, АСНИ и т.д. и т.п.. В таких системах обычно контролируется большое число различных параметров, и для этих параметров может оказаться необходимым изменение единиц измерения.

4.) Кодирование данных. В некоторых ситуациях может понадобиться представлять данные их кодированными значениями. Например, поле 'цвет

ткани', которое в приложении представляется как символьная строка 'красный' в ФБД может храниться в виде десятичной цифры.

5.) Структура хранимых записей.

Таблица 1. 'Бухгалтерия'

Фамилия	Курс, группа	Стипендия
Иванов	IV-1	166р. 98к
Петров	II-2	0

Таблица 2. (диспетчеру, зам. декану)

Фамилия	Оценка пр.1	Оценка пр.2
Иванов	4	5
Петров	3	4

Файл 1.dbf

FAM	KG	S
Иванов	IV-1	166р. 98к
Петров	II-2	0

Файл 2.dbf

FAM	O1	O2
Иванов	4	5
Петров	3	4

Объединение

Файл 3.dbf

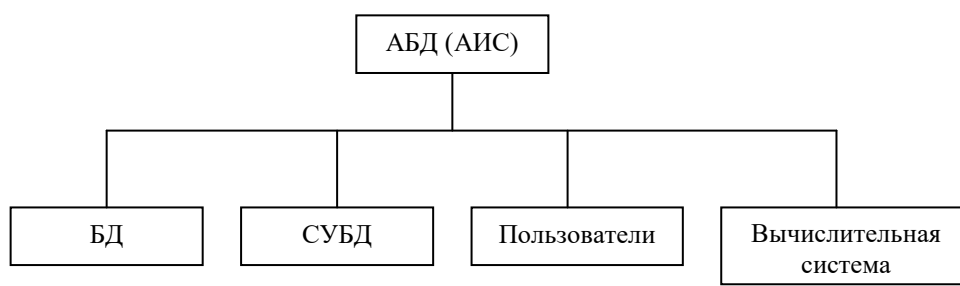
FAM	KG	S	O1	O2
Иванов	IV-1	166р. 98к.	4	5
Петров	II-2	0	3	4

6.) Структура хранимых файлов. Открытый файл физически может храниться в памяти компьютера различными способами. Например, отдельные компоненты могут быть размещены на одном диске, либо на нескольких.

Файл может быть физически упорядочен сначала по одному полю, а затем необходимо его отсортировать по другому хранимому полю. Реально существуют и другие аспекты, которые могут быть изменены в процессе эксплуатации АИС, причем в большинстве случаев эти изменения связаны с изменением или усовершенствованием АИС.

В заключении следует отметить, что независимость данных — понятие относительное, и различные реальные АИС обеспечивают ее в разной степени. Но даже самые современные АИС в этом плане имеют определенные недостатки.

2.2 Банк данных, как автоматизация системы. Компоненты БД



Такая структура действительно соответствует структуре автоматизированных систем. Она является человеко-машинной и предназначена для сбора и обработки данных, используемых в различных сферах человеческой деятельности.

2.2.1. БД

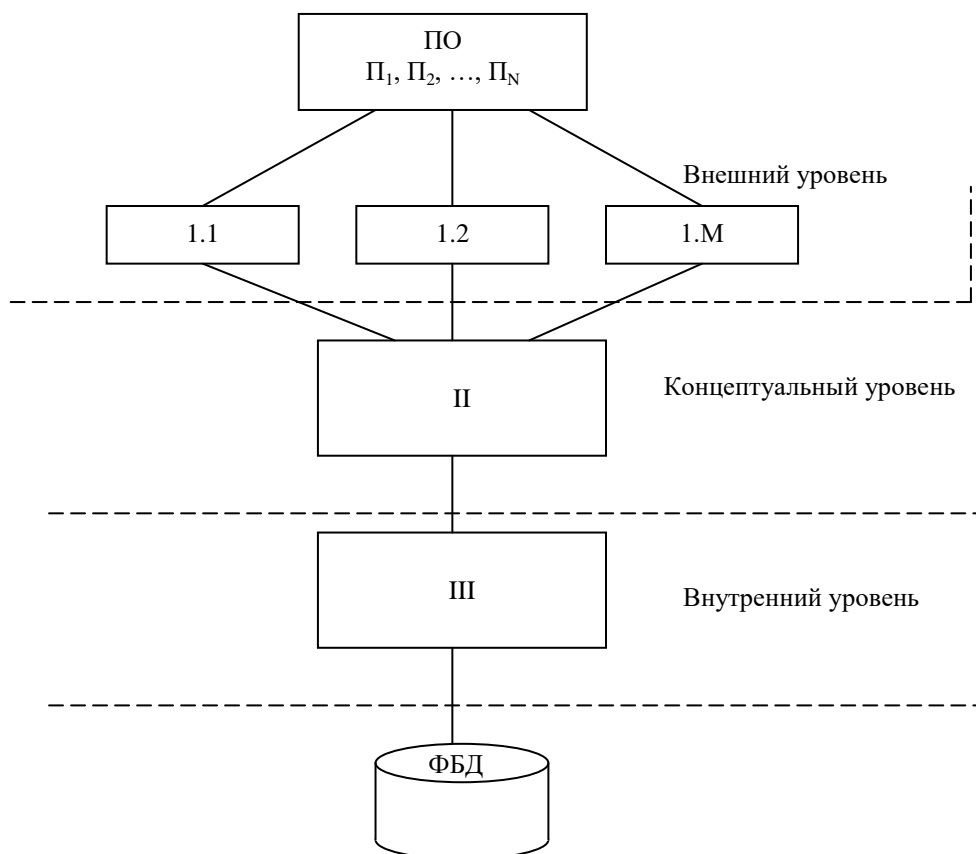
БД предназначена для поддержания адекватной модели предметной области, с целью использования ее в различных приложениях. БД должна

обладать следующими основными качествами:

- 1.) интегрированностью, направленную на решение комплекса задач;
- 2.) структурированностью (модульностью), позволяющей отражать какую-то предметную область в виде адекватной модели;
- 3.) независимостью описания данных прикладных программ;
- 4.) взаимосвязанностью данных, что позволяет используя одно данные получать новые данные на их основе.

В зависимости от того, кто использует или разрабатывает АИС (АБД), возникает как бы уровни абстрагирования или представления о БД.

Наиболее распространена трехуровневая система.



Внешнее представление — это содержимое БД, каким видит его определенный пользователь. Для таких пользователей БД — это хранилище сведений (на стипендии, оценках, ценах,...) о тех объектах, которые понятны специалистам в этой предметной области (ПО).

Второй уровень – концептуальный уровень представления соответствует представлению проектировщика АИС о логической организации данных или о структуре данных. Этот уровень во многом совпадает со внешним представлением, но его отличие состоит в привязке к средствам реализации АИС (СУБД). Описание БД на концептуальном уровне задается на языке описания ФБД, используемого в СУБД.

Пример.

FAM	character	Второй уровень
KG	numeric	
...	...	

Концептуальное представление состоит из множества экземпляров каждого типа концептуальной записи. На каждом уровне идет обобщение индивидуальных представлений пользователя. Определения концептуальной схемы могут включать и другие дополнительные средства — средства обеспечения безопасности, целостности.

Третий уровень – внутренний уровень представления (абстрагирования). Это представление всей БД в виде многих экземпляров каждого типа хранимой записи. Но внутренние представления, такое, как и первые два уровня не связаны с физическим уровнем (физические записи).

Другими словами, внутренне представление предполагает, что память – бесконечное линейное адресное пространство.

Внутренние представления обычно описываются с помощью внутренней схемы, которая определяет различные типы хранимых записей, существующие индексы, способы представления хранимых полей, последовательность хранимых записей и т.д.

{	PREFITY TYPE BYTE (20)
	FAM TYPE BYTE (10)
	KG TYPE BYTE (4)

2.2.2 СУБД

СУБД — специальный пакет программ, посредством которого реализуется централизованное управление БД и обеспечивается доступ к ним.

Упрощенно эти функции реализуются следующим образом:

1. Пользователь выдает запрос на доступ к данным (либо выбором какого-то пункта меню, либо сформировав запрос на языке SQL).
2. СУБД воспринимает этот запрос и анализирует.
3. СУБД просматривает внешнюю схему пользователя, концептуальную схему и внутреннюю схему.
4. После этого СУБД выполняет необходимые операции преобразования данных от одного уровня к другому и в результате инициирует через ОС передачу данных между физической БД и пользователем.

Основные функции СУБД:

1. Обеспечение возможности описания данных: СУБД должна обеспечить определение или описание данных (внешние схемы, концептуальные схемы в виде внутренних схем) и обеспечить преобразование этих описаний в форму соответствующих объектов или другими словами СУБД должна включать в себя языковой процессор для одного языка описания данных.
2. Обработка данных: СУБД должна уметь обрабатывать запросы пользователя на выборку, изменение, удаление существующих записей или на добавление новых данных в базу, или другими словами СУБД должна включать в себя процессор языка манипулирования данными.
3. Обеспечение безопасности и целостности данных: СУБД должна контролировать пользовательские запросы и пресекать попытки нарушения правил безопасности и целостности определенных проектировщиком БД или администратора БД.
4. Обеспечение восстановления и дублирования данных: СУБД или другой связанный с ней компонент должны осуществлять необходимый

контроль за своевременностью процедур дублирования данных и за реализацией процедур восстановления данных при аппаратных сбоях, программных сбоях и т.д.

5. Поддержание словаря данных. Обеспечение функции словаря данных: Этот словарь должен содержать данные о данных (например формы различных схем, перекрестные ссылки, указывающие, какие прикладные программы какую часть БД используют; какие терминалы подключены к системе в данный момент и т.д.).

6. Обеспечение требуемой производительности: СУБД должна выполнять все вышеуказанные функции (особенно манипулирование данными), за временные интервалы, не превышающие требуемых.

2.2.3 Пользователи

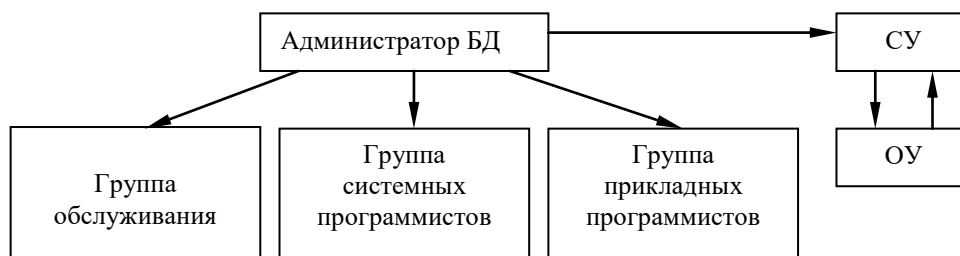
Пользователей АИС (или БД) можно разделить на три группы

1. Прикладные программисты — специалисты, которые создают прикладные программы, взаимодействующие с БД. Прикладные программы могут быть написаны на универсальных языках, либо с использованием языков конкретных СУБД, в любом случае они реализуют какие-то операции, связанные со вставкой, выборкой, удалением данных, либо их обновлением.

2. Конечные пользователи могут получать доступ к данным. Во-первых, используя так называемый интерфейс, основанный на меню и формах. Во-вторых, используя так называемый командный интерфейс, или процессор языка запроса. В этом случае пользователь должен указать команду или выражение высокого уровня (например Select/Insert) обычно с использованием высокого языка (например SQL) и в результате получить те или иные необходимые ему данные. Пользователь может получить доступ к БД используя приложения или программы, разработанные прикладными программистами.

3. Администратор БД — лицо (группа лиц), реализующее управление БД.

Если рассмотреть АСУ, то можно выделить объект управления, то есть БД, и можно выделить управляющий орган, или систему управления, то есть администратора БД.



Основные функции администратора БД:

1. Определение концептуальной схемы — первый этап создания АИС, состоит в том, что специалисты, хорошо знающие эту область, формируют внешние представления или внешние схемы данных, о предметной области. Этот этап обычно называется этапом инфологического проектирования, но и на этом этапе администратор должен уже принимать участие, чтобы иметь возможность создать соответствующую концептуальную схему.

2. Определить внутреннюю схему данных — этот этап, называется этапом физического проектирования.

3. Взаимодействие с пользователями — в функцию администратора БД входит обеспечение пользователей необходимыми данными, консультации по разработке приложений, обеспечение консультаций по техническим и другим вопросам, при возникновении проблем в процессе эксплуатации БД.

4. Определение и реализация мер по обеспечению безопасности и целостности. (например: одна мера по защите данных от некомпетентного использования, а вторая, по обеспечению секретности определенной части данных).

5. Определение процедур резервного копирования и восстановления.

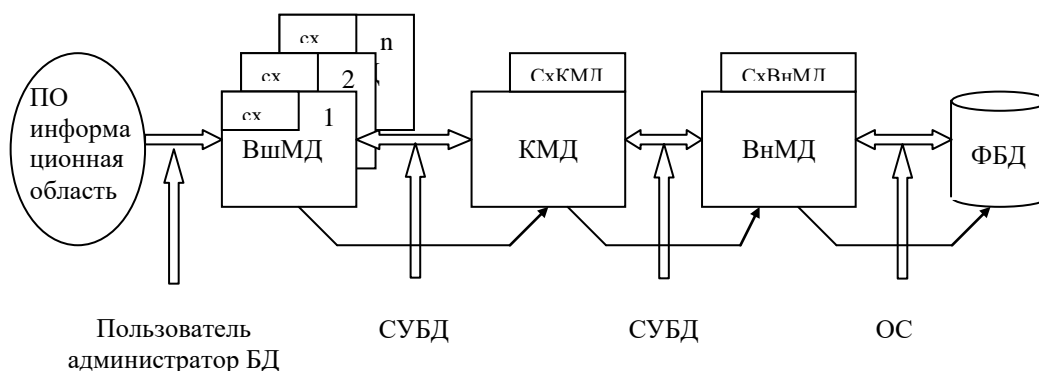
6. Управление производительностью и реагирования на изменяющиеся требования к АИС (АБД).

2.2.4 Архитектура БД.

Для обеспечения независимости ПП от данных, для обеспечения целостности, безопасности данных при построении БД используется т.н. модели данных.

Внешняя модель данных (ВшМД), концептуальная модель данных (КМД) и внутренняя модель данных (ВнМД). Эти модели соответствуют уровням представления или абстрагирования БД. Существует несколько терминов: модель данных (МД), схема МД (СхМД) и отображение МД одной на другую.

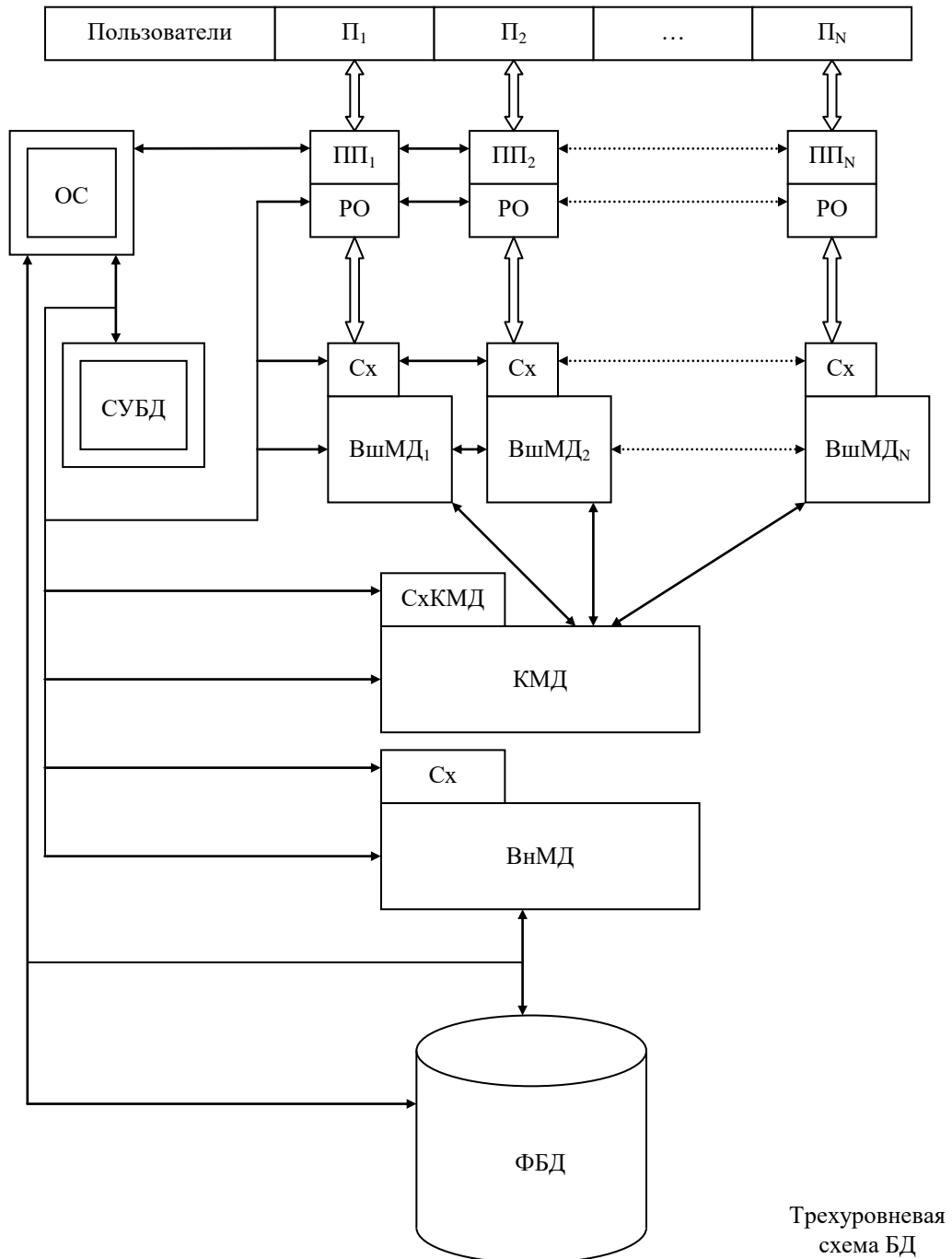
Схема данных — это инструментарий, позволяющий отображать предметную область на определенном уровне абстрагирования.



ВнМД могут быть специальными для каждого пользователя, они обычно имеют несколько схем.

КМД обычно имеют одну схему, в которой отображается структура данных, имена записей, имена полей, форматы полей и т.д.

СУБД реализует обмен данными между рабочими областями ПП и ФБД. Любой запрос ПП, сформулированный на ЯНД, поступает в СУБД. Имея соответствие схемы моделей и описание отображение между моделями, СУБД обращается к материалам доступа ОС для выполнения необходимых действий уже на физическом уровне. Такое взаимодействие между СУБД, ОС и МД может быть представлено в виде трехуровневой архитектуры данных.



Последовательность действий СУБД при формировании записи ВшМД для ПП по запросу следующая:

1. ПП обращается к СУБД для чтения или записи из ФБД в виде ВшМД;
2. СУБД использует схемы ВшМД и КМД и используя отображения ВшМД на КМД, определяет какие записи КМД необходимы для формирования требуемой записи ВшМД;

3. Используя схемы КМД и ВнМД и описания отображения КМД на ВнМД, СУБД определяет, какие хранимые записи необходимы для построения затребованных записей КМД и какая совокупность физических записей необходима для считывания с магнитного диска;

4. СУБД выдает запрос ОС на считывание, в свою буферную область памяти, необходимой записи из ФБД;

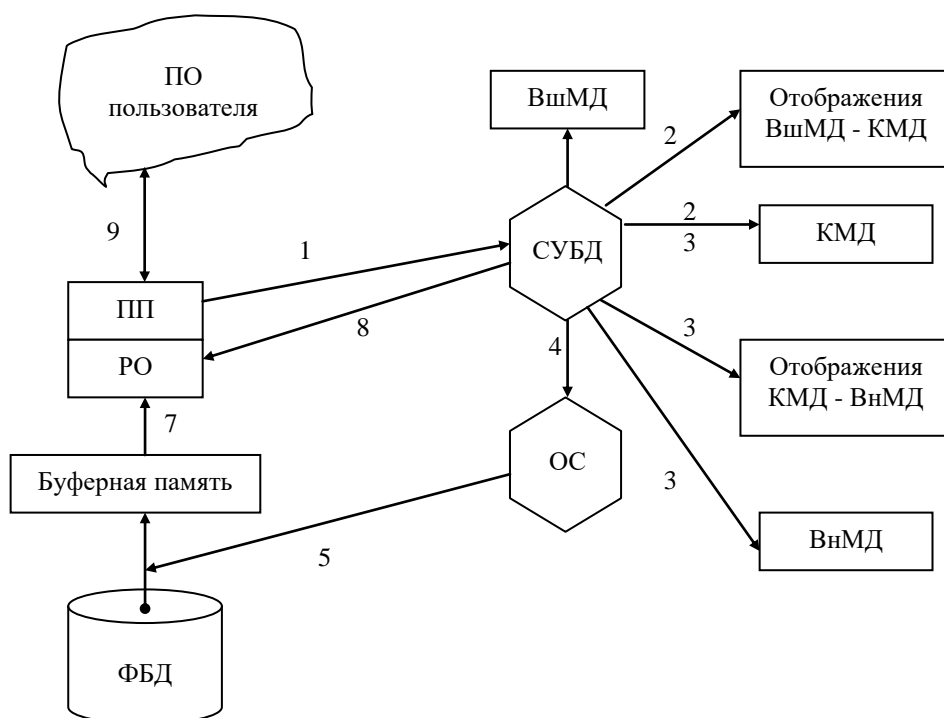
5. ОС, с помощью своих методов доступа, считывает из физической памяти затребованные СУБД физические записи, и помещает их в системный буфер СУБД, кроме того, сообщение ОС о выполнении этого запроса добавляется к сообщениям СУБД, и также помещается в буферную память;

6. На основании имеющихся схем моделей и описаний соответствующих отображений, СУБД формирует в буферной памяти запись ВшМД в таком виде, в каком требует ПП;

7. СУБД пересылает сформулированную запись ВшМД в рабочую область (РО) ввода/вывода ПП;

8. СУБД передает в ПП свои сообщения и сообщения ОС о результатах выполнения запроса;

9. ПП обрабатывает запись, поступившую в РО ввода/вывода, и выдает результат пользователю.

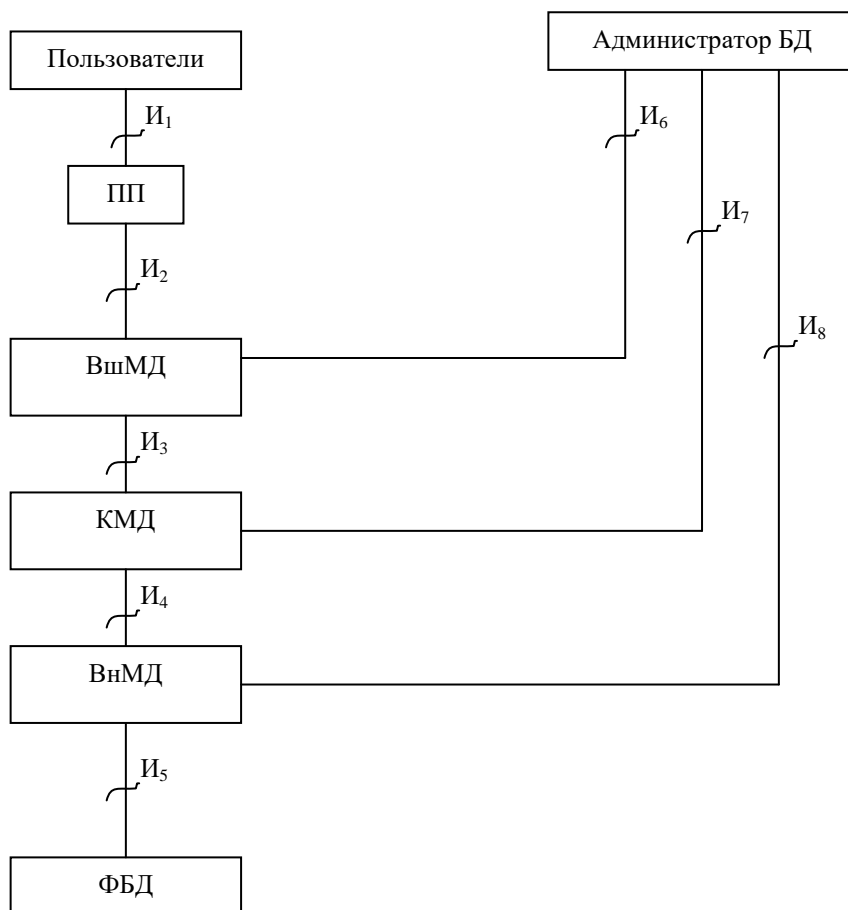


Наличие в БД процессов обмена информацией между пользователем и системой, между администратором БД (и АдБД) и системой, а также между моделями данных различных уровней представления данных требует унификации этих процессов, т. е. разработки соответствующих интерфейсов. Эти интерфейсы разрабатываются с помощью ЯОД и ЯМД на соответствующих уровнях.

В типовом варианте БД обычно используются следующие интерфейсы:

1. Интерфейс пользователя. Используется язык внешнего уровня, при подготовке исходных текстов ПП или формулировке запросов;
2. Имеет место при трансляции ПП или запроса с языка внешнего уровня данные поступают на вход СУБД в объектных кодах;
3. Между КМД и АдБД;
4. Между ВшМД и АдБД;
5. Между ВнМД и АдБД.

Кроме того, для написания и коррекции схем моделей администратор БД может быть обеспечен соответствующими средствами или языками.



При такой архитектуре банк данных обладает высокой способностью адаптации к возможным изменениям, как в прикладных программах (ПП), так и в самих данных, т. е. любые изменения ВшМД и ВнМД схем изолированы друг от друга концептуальной схемой и могут выполняться независимо.

Кроме рассмотренных уже уровней абстрагирования в БД существует ещё один, им предшествующий. Модель этого уровня содержит информацию о ПО в виде, независимом от используемой СУБД (на естественном языке). Это естественный, информационный уровень абстрагирования, связанный с фиксацией и описанием выделенных сведений о ПО. Модель этого уровня называется инфологической моделью ПО.

2.3 Классификация ИС и БД

ИС (БД) могут классифицироваться по различным признакам:

1.) В зависимости от масштаба (ИС рабочей группы, отдела предприятия)

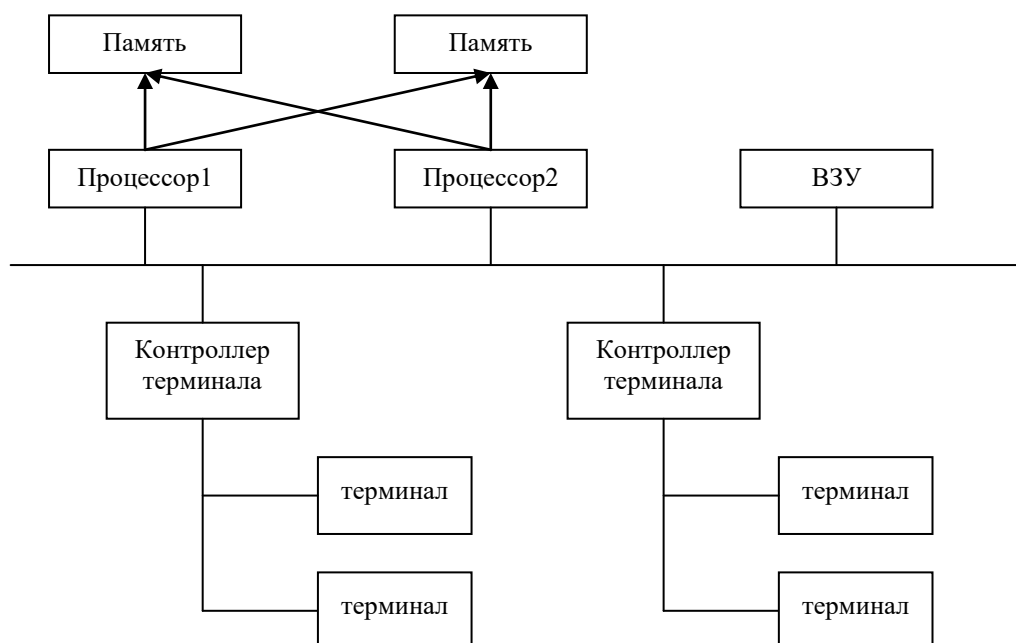
2.) По характеру обработки информации:

- оперативная обработка транзакций;
- системы оперативного анализа;
- системы поддержки принятия решений;
- хранилище данных.

3.) В зависимости от архитектуры:

- ИС с централизованной архитектурой;
- ИС с архитектурой клиент-сервер;
- Корпоративные ИС с использованием интернет-технологий;
- Географически распределенной ИС.

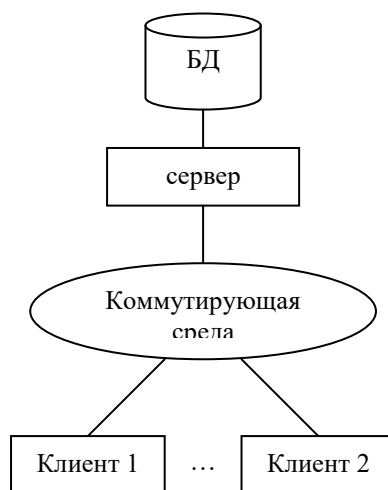
2.3.1 ИС с централизованной архитектурой.



Такая архитектура является традиционной и характеризуется тем, что имеется ЭВМ (центральный узел, HOST), на котором располагается и БД и СУБД и приложения.

2.3.2. ИС с архитектурой клиент-сервер

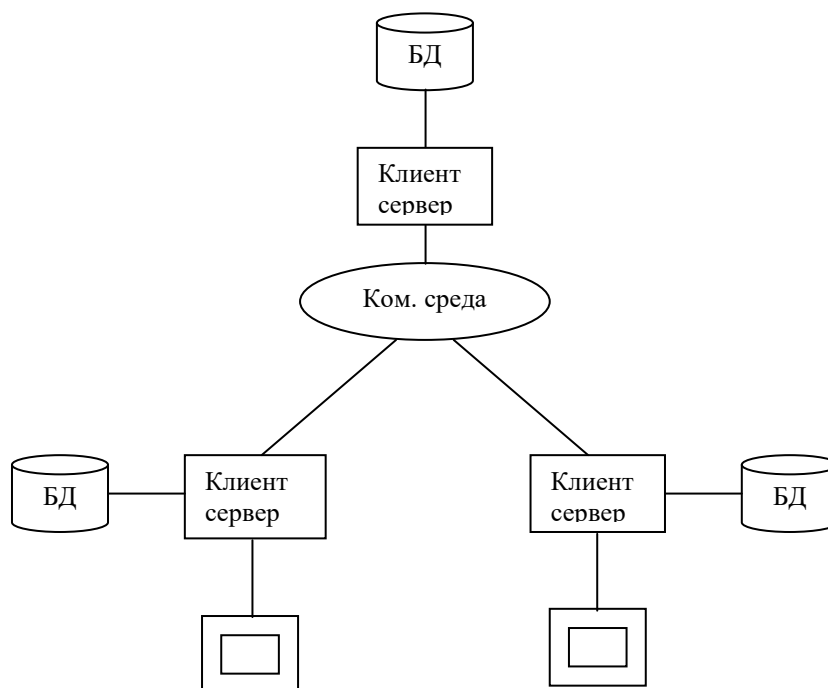
При рассмотрении ИС можно, на более высоком уровне обобщения архитектуры, выделить такие компоненты – сервер (или машины БД, СУБД) и набора клиентов (различные приложения).



В рамках такой архитектуры может быть выделена разновидность — файл-сервер. В этом случае компьютер-сервер и его база (ВЗУ) используется как хранилище данных, а обработка на файл-сервере не производится. В архитектуре клиент-сервер на сервере может осуществляться обработка данных.

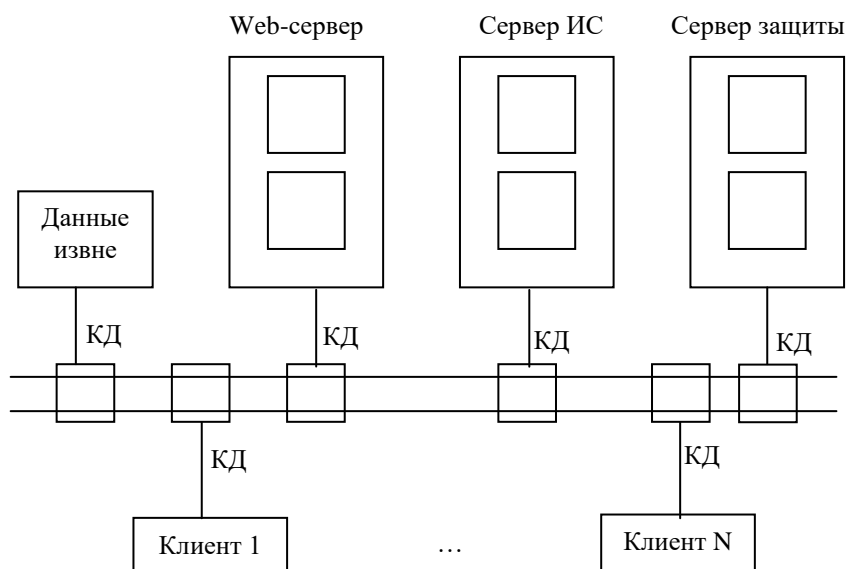
В последнее время в некоторых случаях машины клиентов могут иметь свои собственные сохраняемые данные и машина сервера тоже может иметь свои приложения. То есть, каждый компьютер может выступать и в роли сервера, и в роли клиента. В этом случае говорят, что такая структура поддерживает полную систему БД. В рамках этой архитектуры описывается два варианта доступа:

1. Клиент может получать доступ в каждый момент времени только к одному из серверов;
2. Клиент может получить доступ к любому числу серверов одновременно (распределенная система данных).



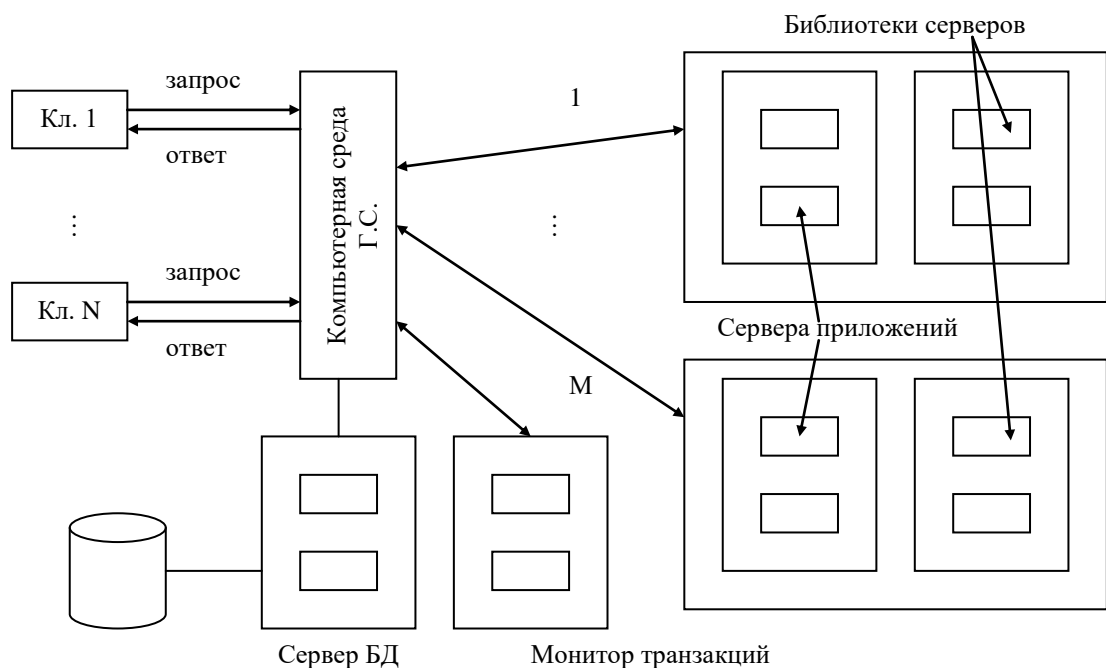
2.3.3 Корпоративные ИС на базе интернет-технологии

В таких ИС обычно осуществляется объединение ИС типа К-С (клиент-сервер) с Web-сервером. Отличительной особенностью таких систем считается возможность хранения не только структурированных, символьных и числовых данных, а также хранение документов, видео, цифровой и звуковой информации.



КД - контроллер доступа.

2.3.4 Архитектура географически распределенных ИС



Особенностью таких систем является то, что клиенты размещены на сравнительно больших расстояниях друг от друга и от сервера БД. Взаимосвязь клиентов с приложениями и БД осуществляется через глобальную сеть. Для определения траектории движения используется так называемый монитор транзакций (логическая единица работы в информационной сети). В таком

варианте сервер БД используется как хранилище. Сервер приложений реализует различные функции по обработке данных. Монитор транзакций отвечает за коммутацию.

3 Данные в ИС

3.1 Жизненный цикл ИС

Жизненный цикл любого объекта, создаваемый человеком, проходит три основных периода:

1. Проектирование, в течении которого создаются модели объекта;
2. Реализация (материализация), в течении которого проект находит свое материальное воплощение;
3. Эксплуатация (объект обеспечивает требуемые функции в реальных условиях).

Жизненный цикл ИС также можно разбить на три основных периода:

- 1) проектирование;
- 2) программная и аппаратная реализация;
- 3) эксплуатация.

Можно выделить следующие этапы проектирования:

1. Анализ ПО;
2. Проектирование концептуальной и инфологической моделей ПО;
3. Выбор СУБД и аппаратных средств;
4. Логическое проектирование, в рамках которого создаются внешние логические модели данных и концептуальные даталогические модели;
5. Физическое проектирование - создание внутренней даталогической модели);
6. Разработка приложений.



Этап инфологического проектирования должен завершиться разработкой инфологической модели, которая по существу является семантической (смысловой) моделью, отражающей смысловое содержание конкретной смысловой области. Для выделяется из воспринимаемой реальности ПО, определяются ее границы, осуществляется абстрагирование несущественных частей и связей в данной ПО с окружающим миром. После этого изучается ПО, накапливаются знания. Эти знания представляются в виде неформализованного описания с использованием специального языка, с применением схем, формул, диаграмм связей и т.д. Выполняется структуризация знаний о ПО: выделяются и классифицируются множество составляющих ПО (объектов), выделяются свойства этих объектов и связи между ними и стандартизируется терминология. Изучается классификация и стандартизация потребностей пользователей (запросов) к инфологической системе.

Третий блок (этап) связан с компоновкой концептуальной инфологической моделью и внешних инфологических моделей. Каждая внешняя инфологическая модель должна отражать отдельные состояния ПО; но это отражение не должно зависеть от методов представления данной конкретной СУБД.

Концептуальная инфологическая модель строится на базе внешних инфологических моделей с учетом их специфики, но обобщает их и взаимосвязывает.

Выбор СУБД аппаратных средств ИС обычно осуществляет группа проектировщиков исходя из опыта эксплуатации уже существующих ИС, функций проектируемой ИС, финансовых возможностей заказчика, а также набора доступных программных и аппаратных средств. В настоящее время существует широкий набор как СУБД, применимый для тех или иных ИС (DBASE, FOXBASE, FOXPRO, CLIPPER, CLARION, ACCESS, PARADOX...).

Существует СУБД для промышленных ИС на базе архитектуры клиент-сервер (ORACLE, INFORMIX, PROGRESS, SYBASE, MS SQL SERVER).

Задача логического этапа проектирования – организация данных, выделенных на предыдущем этапе проектирования в форму, выработанную конкретным выбором СУБД. Требуется разработать схему концептуальной датологической модели и схемы внешних датологических моделей данных, пользуясь только теми типами моделей данных и их особенностями, которые поддерживаются этой СУБД.

Задача физического этапа проектирования – выбор рациональной структуры хранения данных и метода доступа к ним, исходя из арсенала методов и средств, который представляется разработчику системой уравнения БД.

Задача 7-го этапа – разработать ПО для всех приложений.

Этап сопровождения (эксплуатации). При внедрении ИС схемы наполняются конкретными данными и начинается эксплуатация. При этом обычно возникают неувязки, неправильные действия персонала, выявляются ошибки, недочеты в проекте.

3.1.1 Анализ предметной области (ПО)

БД – модель ПО, в которой должны быть отражены только те факты, которые необходимы для ее функционирования. При проектировании БД проектировщик должен выделить и описать эти ожидаемые факты и тем самым будет отчерчено граница ПО. Дальнейшее изучение ПО должно обеспечить не только выделение объектов, но и описание их свойств, связей друг с другом, то есть для каждого типа объектов фиксируется совокупность свойств, посредством которых будут описываться конкретные объекты этого типа в БД, а так же виды взаимосвязей (отношения) между этими объектами.

В настоящее время наиболее распространены, известны три подхода для проведения анализа:

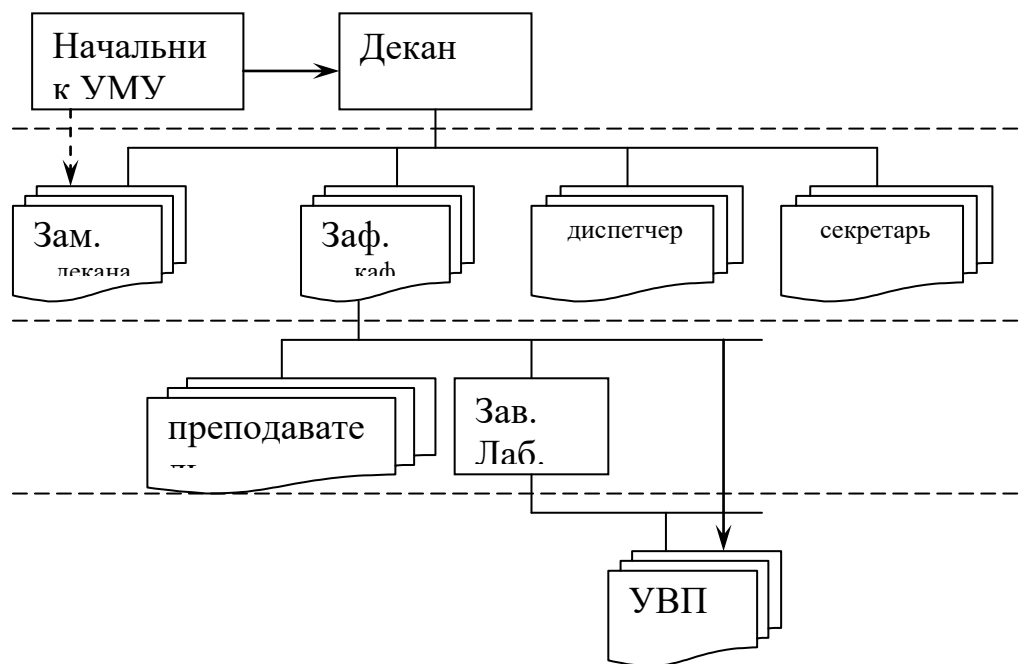
1. Традиционная схема, ориентированная на использование различных нормативных документов и ГОСТов.
2. Схема структурного анализа.
3. Схема, ориентированная на потоки данных.

3.1.2 Традиционная схема анализа ПО.

Традиционная схема предполагает, что в результате проведения этого этапа формируются документы включающие:

- схемы организационной структуры ПО;
- функциональную схему;
- схему информационных потоков.

Исходные данные: результаты анкетирования персонала, должностные инструкции, наблюдение и описание компонент объекта разработчика, собранные документы.



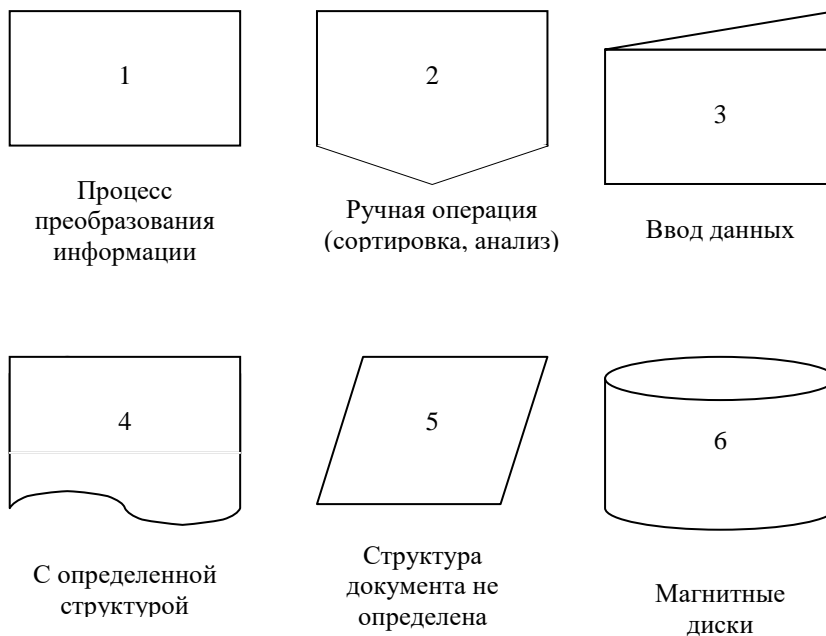
Функциональные схемы составляются для отражения каждого из процессов и указания подразделения или объекта, в котором выполняются функции, относящиеся к рассматриваемому процессу. При составлении функциональной схемы определяется перечень элементарных действий и их информационных взаимосвязей, а так же указываются внешние объекты, взаимодействующие с подразделением и участвующие в этом процессе.

Функция представляется в виде прямоугольника, имя функции указывает на выполняемое действие.

Пример. Функциональная схема по реализации процесса назначения стипендии студентам и схема информационных потоков:



Схема информационных потоков данных представляется в соответствии с ГОСТами:





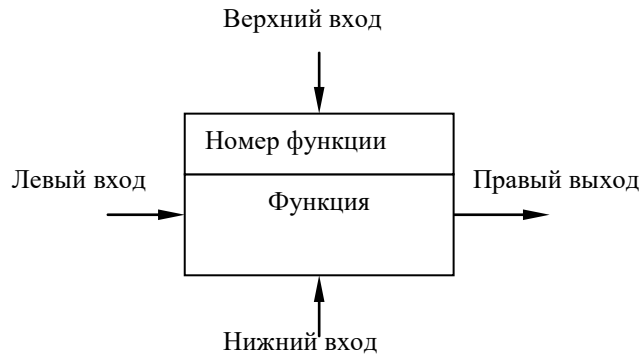
3.1.3. Схемы структурного анализа.

Этот подход был разработан в рамках Минобороны США.

Такой подход предполагает построение функциональной модели на базе элементов, в которых определены разновидности входов и выходов в зависимости от их положения по отношению к элементу.

Т.е. каждый элемент в виде прямоугольника отображает выполняемую функцию

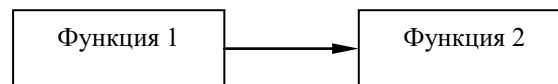
- левый вход – поступающая входная информация;
- верхний вход – операция на ограничение реализации функции;
- нижний вход – механизм (субъект) реализации функции;
- правый выход – выходная (переработанная) информация.



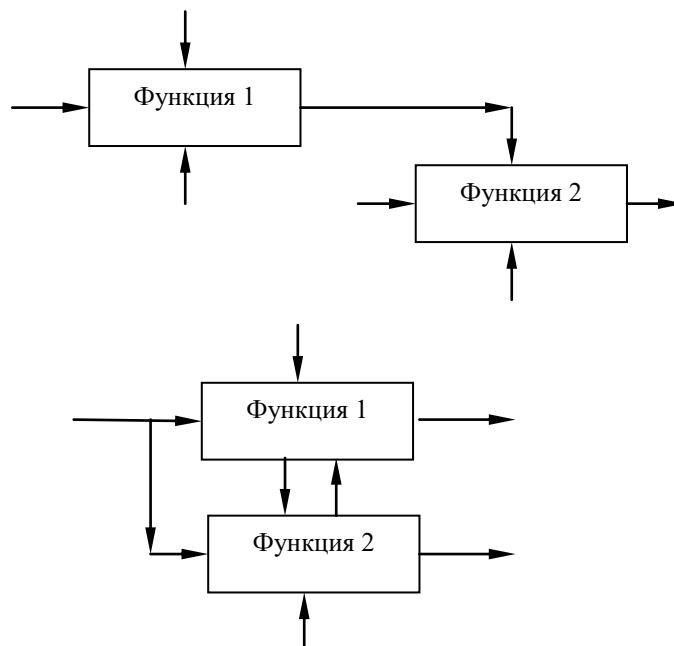
Далее эти элементы объединяются друг с другом. При этом между ними могут быть различные связи:

1. последовательный вид связи;
2. функциональная связь;
3. коммуникационная связь;

Последовательная связь: соответствует ситуации, когда выход функции 1 является входом для функции 2.



Функциональная связь: выход функции 1 задает ограничения для функции 2.



Коммуникационная связь: исходные данные параллельно поступают на вход функции 1 и функции 2.

3.1.4 Схемы, ориентированные на потоки данных

В рамках таких схем организационная структура не составляется, а первой (исходной) схемой является схема, отражающая иерархию выполняемых функций.

Обычно выделяются следующие функции:

- преобразование данных;
- ввод данных;
- контроль данных.



На основе приведенного перечня функций, результата анализа документооборота, нормативных документов составляется диаграмма потоков данных, основными компонентами которых являются:

- внешние объекты;
- процессы;
- хранилища данных;
- потоки данных (информации).

Внешний объект – это объект материальной, энергетической или информационной природы, который находится за пределами разрабатываемой ИС.

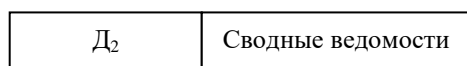
Процесс – это совокупность действий выполняемых ИС и направленных на преобразование данных.

Хранилище данных – это объект, который предназначен для долговременного хранения информации и к которому возможен доступ в произвольные моменты времени.

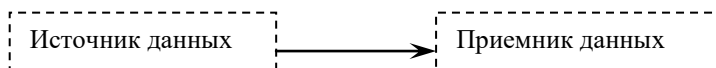
Поток данных предназначен для отображения передачи данных между процессами и хранилищами, а так же для отображения информационных связей с внешними объектами.

Указанные компоненты отображаются на схемах следующим образом:

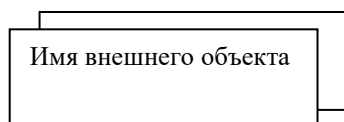
1.Хранилище: (прямоугольник)



2. Поток:

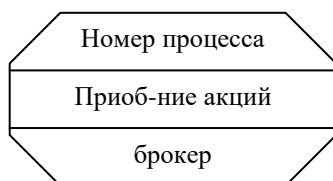


3. Внешний объект:

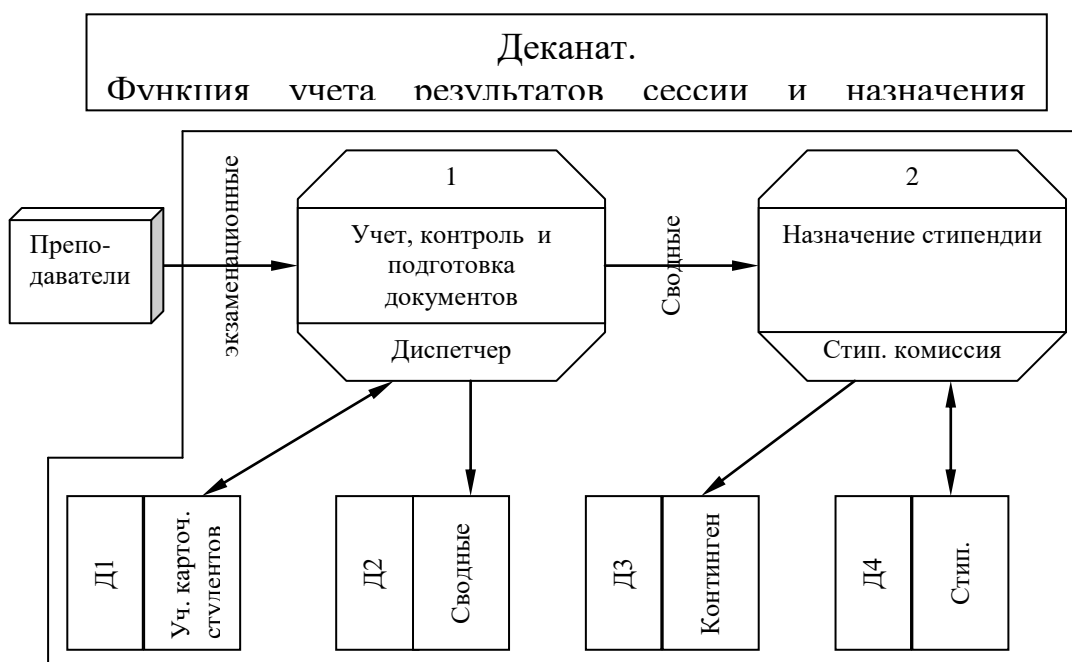


(двойной прямоугольник)

4. Процесс:



(многоугольник)



3.1.5 Проектирование концептуальной инфологической модели

Концептуальные инфологические модели (КИМ) строятся на основе результатов анализа предметной области (ПО).

В некоторых непонятных человеку концепциях, полностью независимые от параметров среды хранения данных. Существует несколько подходов к построению КИМ: графовые модели, информационные сети, модели сущность-связь.

Модель сущность-связь формальная модель предметной области, которая позволяет описывать объекты ПО, взаимосвязь объектов, и может быть использована с будущими потребителями (пользователями) ИС, а также для проектирования БД на даталогическом уровне. Одним из вариантов модели сущность-связь, является представление ПО в виде так называемых 'Е-R' диаграмм.

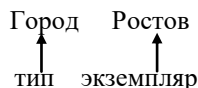
Общими для всех подходов является использование трех основных конструктивных элементов для представления состава ПО — это сущность (объект), атрибуты (свойства), связи (отношения).

Определение: сущность — собирательное понятие, или абстракция

реально существующего объекта, процесса или явления, о котором надо хранить информацию в БД.

Необходимо различать понятия: тип сущности и экземпляр сущности.

Тип сущности определяет набор (класс) однотипных объектов, а экземпляр сущности — конкретный объект из данного набора.



Атрибут — поименованная характеристика сущности, которая принимает значения из некоторого множества значений. В моделях атрибут выступает в качестве средства, с помощью которого моделируется свойство сущностей. Необходимо различать такие понятия, как: тип атрибута, и экземпляр атрибута. Тип атрибута определяет набор однородных атрибутов, а экземпляр атрибута — конкретное значение атрибута или свойства.

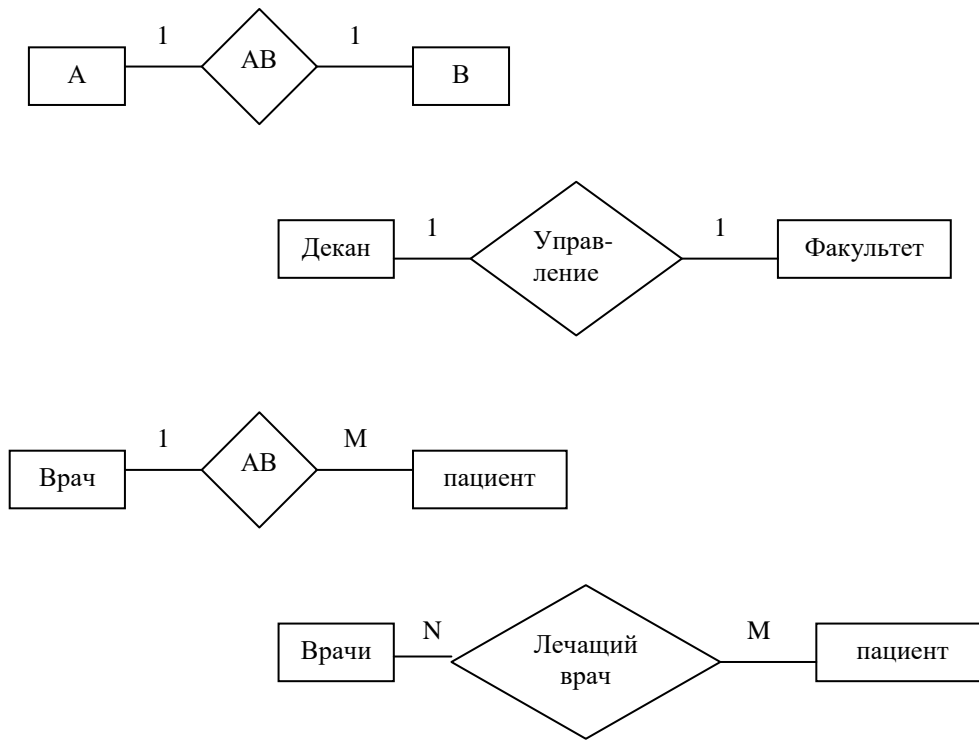
Название города	Статус города	Численность населения
Ростов	Обл. центр	1 млн чел
Новочеркасск	Гор. Обл. значения	180 тыс чел

Ключевой атрибут — минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

Связь — средство модели, с помощью которого отображается взаимное влияние сущностей друг на друга.

Различают понятия: тип связи и экземпляр связи. Понятие 'тип связи' используется для отражения связи между типами связи. Экземпляр связи рассматриваемого типа существует между конкретными экземплярами рассмотренного типа сущности.

При построении E-R диаграмм сущности изображаются в виде прямоугольников, атрибуты — в виде овалов, а связи — в виде ромбом или шестиугольников. При построении E-R диаграмм между отдельными сущностями могут быть различные виды связей.



В том случае, если объект сложен, то Е-Р диаграммы получаются весьма сложными и не очень наглядными. В таких случаях применяется менее наглядной, он более содержательный язык моделирования, в котором сущности и связи представляются выражениями типа:

Сущность I [AT₁, AT₂...]

Связь N [СУЩ₁, СУЩ₂...][AT₁, AT₂...]

Лучше всего ПО разбивать на отдельные объекты и для них создавать КИМ в виде Е-Р диаграмм.

При моделировании КМД ПО, целесообразно представлять ее в виде ряда так называемых локальных представлений и моделировать каждое локальное представление, а затем объединять их в единую КИМ.

3.2 Реляционные модели данных

3.2.1 Модели данных

Проектирование логической структуры (логическое проектирование) БД производится на основе КИМ, и выбранной СУБД. Каждая СУБД поддерживает ту или иную предметную область, но включает в себя средства ориентированные на определенные структуры данных, этот инструментарий называется моделью данных. Модель определяется тремя основными компонентами:

Допустимая организация данных;

Множество операций, допустимых над объектами моделей данных;

Ограничения целостности.

Каждая ЭВМ обладает своей моделью данных (физическая, основная), это допустимые для ЭВМ:

форматы данных;

операции, выполняемые над данными.

С помощью физической модели данных строят более сложные модели данных, т.е. фактически делается переход к некоторой абстрактной ЭВМ, обладающей более удобной моделью данных для решения задач. Пример АЭВМ: трансляторы, интерпретаторы и т.д. Каждый такой язык имеет свою модель данных, которая не зависит от машины и управляет моделированием реальных ситуаций. К примеру, модель данных ПАСКАЛЯ содержит массив переменных различных типов и содержит набор операторов и процедур, гораздо более широкий, чем набор команд процессора. Все операторы любого алгоритмического языка могут быть разделены:

Декларативные операторы (операторы описания);

Процедурные операторы (операторы манипулирования).

Множество допустимых состояний, исходных, промежуточных и выходных данных в программе на алгоритмическом языке, определяется

декларативными операторами. А изменение состояния данных в программе – процедурными операторами.

То есть совокупность декларативных и процедурных операторов и определяет модель данных для этого алгоритмического языка. Все вышесказанное относится к модели данных, поддерживаемой конкретной СУБД, т.е. с точки зрения прикладной программы, СУБД может рассматриваться как некий алгоритмический язык, на котором программируются задачи ИОСУ. Отличие от обычного алгоритмического языка, заключается в том, что декларативные и процедурные операции разделены и оформлены в виде отдельных языков описания данных и манипулирования данными (ЯОД и ЯМД соответственно), и эти языки как раз и определяют модель данных, поддерживаемых конкретной СУБД. Причем модель данных в такой интерпретации понимается как:

Совокупность структур данных;

Совокупность допустимых операций;

Совокупность ограничений.

3.2.2 Структуры данных

Структуризация данных базируется на концепции агрегации и обобщения данных. Например, в файловых системах реализована модель типа «плоский файл». Она использует четыре компоненты логических структур данных:

поле – наименьшая поименованная единица данных;

запись – поименованная совокупность полей;

файл - поименованная совокупность записей одного типа;

набор файлов - поименованная совокупность файлов.

В такой модели агрегация используется для композиции полей в запись, а обобщение – для представления множества экземпляров одного типа, одной общей строкой более высокого уровня — файла.

Пример. Необходимо представить в виде данных информацию о сотрудниках какой либо организации. Пусть отдельные свойства моделируются следующими типами полей: FA, BO, LE. Описание сотрудника в целом может осуществляться с использованием схемы записи типа: Сотрудник(FA, BO, LE). То есть запись, полученная путем агрегации таких полей как: FA, BO, LE. Обобщение всех экземпляров записей типа Сотрудник, может быть осуществлено в файлах типа Сотрудники.

То есть представляется совокупность описаний всех сотрудников данной организации с помощью одного экземпляра файла. Другая форма представления данных, являющаяся удобной с точки зрения понимания и интерпретации — табличная форма. Таблица — совокупное описание экземпляров сущности некоторого типа.

Пример использования табличной формы для описания экземпляра сущности.

№ пр.	Название проекта	Дата выполнения
008	Проверка устройств обмена данными	10.07.99
012	Устройство тестирования блоков ЭВМ	20.03.2000
...
082	Моделирование работы АЛУ	30.07.2000

Таблицы могут использоваться и для описания экземпляра связи некоторого типа между сущностями. При определенных допущениях может использоваться математическое понятие «отношения» для формального описания таблицы.

№ отдела	№ проекта
01	008
03	082
...	...
09	012

Это предложил ученый Е. Кодд. При таком подходе, для обработки данных можно использовать аппарат для теории множеств, и представление данных сводится к представлению двумерных таблиц в виде отношений. При таком подходе сущность представляется агрегатом атрибутов, а связь — агрегатом идентифицирующих атрибутов. То есть в этом случае столбцы таблицы используются для задания атрибутов, а экземпляры сущностей представляются агрегатами значений соответствующих атрибутов.

3.2.3 Основные операции.

Динамические свойства модели данных определяются множеством операций, которые задают действия над содержимым БД (язык описания моделей данных – ЯМД). Реализация любой операции над данными:

Селекция данных (может быть осуществлена с использованием локальной позиции, с использованием значения, с использованием связи между данными);

Действия над выделенными данными (может быть чтение данного из базы, включение данного в базу, удаление, редактирование и т. д.).

По характеру способа получения результата, различают навигационные и спецификационные операции:

- навигационные операции: результат операции получен путем прохождения по связям, имеющим место в БД;
- спецификационные операции: когда в операции определяются только требования к результату, но не задается способ получения результата.

Есть еще обобщенные операции: последовательность операций позволяющих организовать последовательности действий. Процедуры, т.е. обобщенные операции, мощные гибкие средства, позволяющие существенно расширить динамические свойства БД.

3.2.4 Ограничения целостности.

Логические ограничения, которые накладываются на данные, с целью защиты их от санкционированного, но не компетентного пользователя, называется ограничением целостности. Термин - целостность – используется в смысле точности (корректности, безопасности) данных базы. Ограничения могут разделяться: на внутренние и явные ограничения целостности.

Внутренние ограничения целостности: обусловлены тем, что модель данных уже определяет допустимую структуру данных, которые поддерживаются соответствующей СУБД, следовательно, если вводимые данные не соответствуют допустимой структуре, то система должна отвергать вводимые данные. Например, если запись может содержать не более 20 полей, то попытка ввести данные с большим числом полей, будет отвергнута.

Явные ограничения целостности: ограничения, явно специфицированные в БД с помощью специальных конструкций языка описания данных (ЯОД).

Пример: БД "служащие":

Табличный №	Фамилия	Имя	...	Должность
-------------	---------	-----	-----	-----------

Если табличный номер — ключевое поле, то попытка ввода записи с уже имеющимся значением в поле Таб. №, будет отвергнута. Ограничения целостности классифицируются также:

Ограничения домена;

Ограничения атрибута;

Ограничения отношения;

Ограничения БД.

Для организации ограничений целостности домена задаются дополнительные значения для данного домена (пример: для поля должность можно задать список должностей, попытка вписать должность не входящую в список должна быть отвергнута).

Ограничения целостности атрибута – взаимосвязано с ограничениями целостности домена.

Ограничения целостности отношения – формируются из целого ряда ограничений, в виде единого комплекса требований.

К примеру, домен "должность" должен содержать в себе все атрибуты должностей: Ректор, Декан, Зав. Кафедрой, Доцент, Ассистент.

В ограничении целостности отношений задается набор или композиция требований по отношению к различным атрибутам, или по отношению к сочетанию атрибута и система должна этот комплекс требований проверять для конкретных данных. Система должна проверять эти данные по конкретной записи (объекту).

В завершении четвертого раздела, следует отметить, что в настоящее время разработано множество СУБД, ориентированных на разные типы данных. Первая группа систем – это система инвертированных списков, иерархических и сетевых. Вторая группа – реляционные модели данных. Третья группа – постреляционные системы: дедуктивные, экспертные, объектно-ориентированные, семантические и универсальные СУБД.

3.3 Языки запросов для работы с БД.

3.3.1. Краткая характеристика языков.

Основными языками для работы с РБД являются ISBL, SQL, QBE.

1. ISBL – язык реляционной алгебры, т.е. в нём определяется последовательность операций, которые необходимо выполнить над отношениями для получения результата.

2. Языки QBE и SQL относятся к языкам реляционного исчисления.

В QBE в качестве объектов исчисления используются столбцы, а в SQL – кортежи. В качестве наиболее распространённого (основного) языка используется SQL (Structured Query Language) – язык структурированных запросов данных. Язык SQL поддерживают практически все коммерческие СУБД: Orakl, Sybase, DB2, MS SQL и т.д.

Необходимо отметить, что реализация языка SQL, а также типы используемых данных, несмотря на наличие стандартов, различаются в разных СУБД. Это создаёт определённые сложности при переходе от одной СУБД на другую или при переносе приложений.

3.3.2 Язык SQL

Этот язык был разработан в компании IBM Research в начале 70-х годов и затем модернизировался.

SQL1–определён в стандарте 1989 г.

SQL2–определён в стандарте 1992 г. (SQL92)

SQL3–в стадии разработки.

Ни один из коммерческих продуктов не поддерживает в полной мере стандарт SQL2. Фактически, каждый коммерческий продукт представляет собой “надмножество подмножества” языка SQL2.

Продукт IBM DB2, не поддерживает всех средств SQL2, касающихся целостности, но он превосходит стандарт в отношении правил, связанных обновлением представлений.

Замечание: в SQL2 используются термины ТАБЛИЦА, СТРОКА, СТОЛБЕЦ вместо реляционных терминов ОТНОШЕНИЕ, КОРТЕЖ, АТРИБУТ.

Основные компоненты языка SQL:

Язык SQL включает в себя (ЯОД) язык определения данных (DDL–Data Definition Language) и язык обработки данных (DML–Data Manipulation Language) – ЯМД.

ЯМД SQL можно использовать и на внешнем и на концептуальном уровне, при описании проекта АБД и систем ИО.

3.3.3 Определение данных

Основные операторы языка определения данных в SQL:

Создание CREATE DOMAIN CREATE TABLE

Изменение ALTER DOMAIN ALTER TABLE

Уничтожение DROP DOMAIN DROP TABLE

Домены в SQL используются для обеспечения возможности один раз определить элементарную спецификацию типа данных, а затем использовать её одновременно для нескольких столбцов в нескольких базовых таблицах.

3.3.4 Синтаксис оператора создания домена

CREATE DOMAIN имя_домена тип_данных [массив_уточненный]

В языке SQL поддерживаются скалярные типы данных

CHARACTER (n) INTEGER DATE – дата

BIT (n) FLOAT (p) TIME – время

NUMERIC (p,g) TIME STAMP и т.д.

DECIMAL (p,q) INTERVAL–временный интервал

VAR CHAR (n) – текстовые данные неизменной длины.

В списках уточнения могут указываться значения включаемые в столбец из домена по умолчанию. Могут указываться определения (описания) ограничений, применяемых к каждому столбцу, создаваемых в этом домене.

Создается БД:

S (поставщики)

NS	NAM E	STATU S	CITY (место размещения)
S1	Смит	20	Лондон
S2	Джон	10	Париж
S3	Том	30	Париж
S4	Кларк	20	Лондон
S5	Адам	30	Афины

Р (детали)

NP	PNAME (название)	COLOR (цвет)	WEIGHT (вес)	CITY (место хранения)
P1	Nut (гайка)	Red (красный)	12	Лондон
P2	Болт	Green (зелёный)	17	Париж
P3	Шуруп	Blue (голубой)	17	Рим
P4	Шуруп	Red (красный)	14	Лондон
P5	Палец	Blue (голубой)	12	Париж

SP (поставки)

NS	NP	QTY (шт)
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

```
CREATE DOMAIN PNAME CHAR (6);
CREATE DOMAIN NS CHAR (5);
CREATE DOMAIN NAME CHAR (20);
CREATE DOMAIN STATUS NUMERIC (5);
CREATE DOMAIN CITY CHAR (15);
CREATE DOMAIN NP CHAR (6);
CREATE DOMAIN COLOR CHAR (6);
CREATE DOMAIN WEIGHT NUMERIC (5);
CREATE DOMAIN QTY NUMERIC (6);
```

3.3.5 Синтаксис оператора определения таблицы.


```

CREATE TABLE <имя таблицы>
[<имя столбца> <тип данных> [<размер>],
-----
-----]);
или в более общем виде
CREATE TABLE table (base-table-element-commalist);

```

Где

1.table – имя таблицы;

2.base-table-element (список элементов таблицы разделённых запятыми)

– каждый элемент таблицы является либо определением (описанием) столбца либо определением ограничения таблицы (base-table-constraint-definition)

2.1. Каждое определение столбца в свою очередь выглядит следующим образом:

column representation [default-definition] <имя столбца> <тип данных или домен> [размер] [размер по умолчанию].

Здесь column означает название столбца;

representation – указывает необходимый тип данных или домен;

default-definition – необязательное определение по умолчанию, т.е. указывает значение по умолчанию для этого столбца. Если для данного столбца не определено его собственное явное значение по умолчанию, а также нет такового наследуемого из домена, то предполагается, что значение его по умолчанию „NULL”.

Существует три (3) вида определений ограничений таблицы:

1. Определение потенциального ключа;
2. Определение внешнего ключа;
3. Определение проверочного “условия”.

Потенциальные ключи.

Определение потенциального ключа записывается в виде:

UNIQUE (column-commalist) или UNIQUE (список имён столбцов разделённых запятыми)

В общих случаях список имён столбцов (column-commalist) не должен быть пустым. Для данной базовой таблицы должно быть не более одного первичного ключа (primary key) и любое количество альтернативных (вторичных) ключей (UNIQUE).

Для первичного ключа для каждого из указанных столбцов подразумевается значение NOT NULL, даже если это не указано явно.

Внешние ключи.

Определение внешнего ключа записывается следующим образом:

FOREIGN KEY (<список имён столбцов, разделённых записями>)

REFERENCES <имя таблицы> (<имя столбца1>, <имя столбца 2>, ...)

Проверочные условия.

Могут также задаваться в списке элементов в качестве ограничений

CHECK (выражение условия для ограничения)

Пример создания таблицы с помощью оператора:

CREATE TABLE SP (NS, NP, QTY, PRIMARY, KEY(NS, NP), FOREIGN KEY(NS), REFERENCES S, FOREIGN KEY (NP), REFERENCES P, CHECK (QTY>0 AND QTY<501));

SP (поставки)

NS	NP	QTY (шт)
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400