Тема: Информационный процесс накопления данных

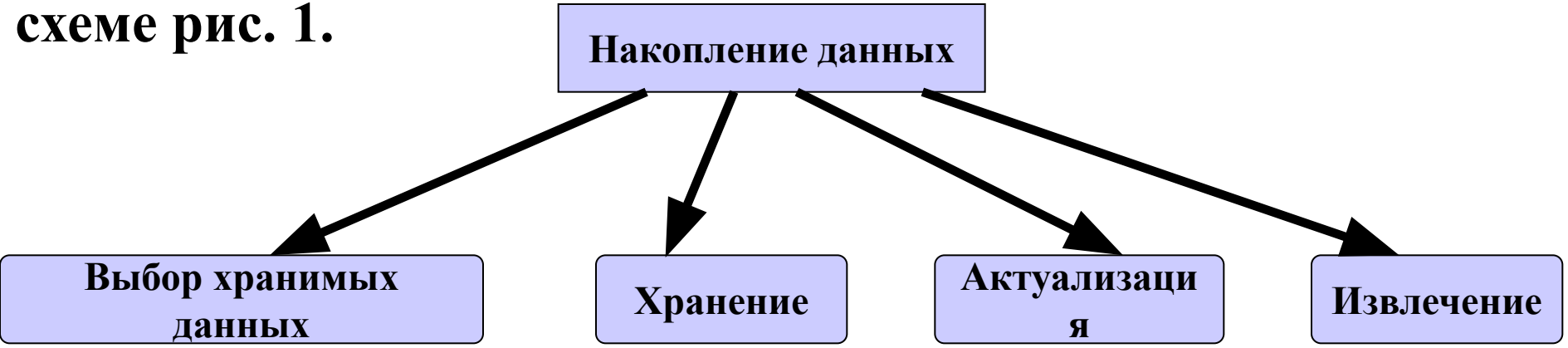
План лекции:

1. Информационная сущность накопления данных.
2. Выбор хранимых данных.
3. Базы Данных.
4. Реляционная модель баз данных.
5. Научно-методический аппарат описания экономической информации в реляционной БД.
6. Объектная модель БД.
7. Программно-аппаратный уровень процесса накопления данных (СУБД).

1. Информационная сущность накопления данных.

Назначение технологического процесса накопления данных состоит в создании, хранении и поддержании в актуальном состоянии информационного фонда, необходимого для выполнения функциональных задач системы управления, для которой построен контур информационной технологии.

Процессы и операции, выполняемые при протекании процесса накопления данных, приведены на структурной схеме рис. 1.



Информационный фонд систем управления должен формироваться на основе принципов необходимой полноты и минимальной избыточности хранимой информации. Эти принципы реализуются процедурой выбора хранимых данных, в процессе выполнения которой производится анализ циркулирующих в системе данных и на основе их группировки на входные, промежуточные и выходные, определяется состав хранимых данных.

Входные данные - это данные, получаемые из первичной информации и создающие информационный образ предметной области. Они подлежат хранению в первую очередь.

Предметной областью называются элементы материальной системы, информация о которых храниться и обрабатывается в экономической информационной системе.

Информационным обеспечением всей предметной области экономического объекта служит информационная база ЭИС.

Промежуточные данные - это данные, формирующиеся из других данных при алгоритмических преобразованиях. Как правило, они не хранятся, но накладывают ограничения на емкость оперативной памяти компьютера.

Выходные данные являются результатом обработки первичных (входных) данных по соответствующей модели и входят в состав управляющего информационного потока своего уровня и подлежат хранению в определенном временном интервале.

Вообще, данные имеют свой жизненный цикл существования, который фактически и отображается в процедурах процесса накопления. Структурная схема жизненного цикла данных приведена на рис. 2.

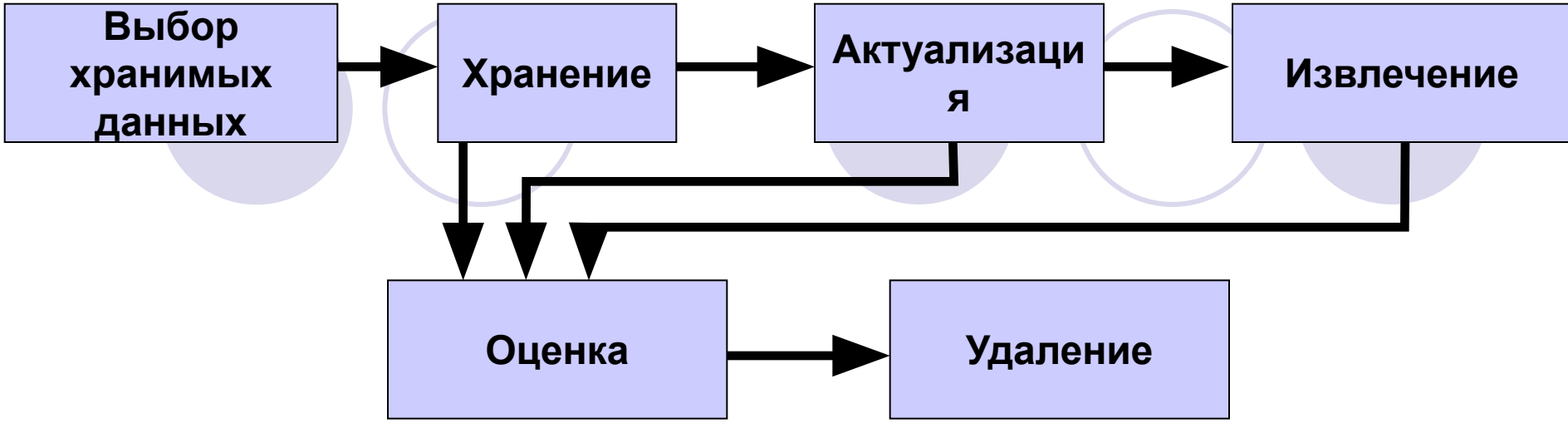


Рис. 2. Структурная схема жизненного цикла существования данных

Из этой схемы следует, что процедуры хранения, актуализации и извлечения данных должны периодически сопровождаться оценкой необходимости их хранения, так как данные подвержены старению. Устаревшие данные должны быть удалены.

Процедура хранения состоит в том, чтобы сформировать и поддерживать структуру хранения данных в памяти ЭВМ.

Современные структуры хранения данных должны быть независимы от программ, использующих эти данные и реализовывать вышеуказанные принципы (полнота и минимальная избыточность). Такие структуры получили название баз данных (БД).

Осуществление процедур создания структуры хранения (базы данных), актуализация, извлечение и удаление данных производится с помощью специальных программ, называемых системами управления базами данных (СУБД).

Процедура актуализации данных позволяет изменить значения данных, записанных в базе, либо дополнить определенный раздел, группу данных. Устаревшие данные могут быть удалены с помощью соответствующей операции.

Процедура извлечения данных необходима для пересылки из базы данных требующихся данных либо для преобразования, либо для отображения, либо для передачи по вычислительной сети.

При выполнении процедур актуализации и извлечения обязательно выполняются операции *поиска* данных по заданным признакам и их *сортировки*, состоящие в изменении порядка расположения данных при хранении или извлечении.

2. Выбор хранимых данных.

Информационный фонд системы управления должен обеспечивать получение выходных наборов данных из входных с помощью алгоритмов обработки и корректировки данных. Это возможно, если создана инфологическая модель предметной области, которая вместе с наборами хранимых данных и алгоритмами их обработки позволяет построить каноническую модель (схему) информационной базы, а затем перейти к логической схеме и, далее, к физическому уровню реализации.

Инфологической (концептуальной) моделью предметной области называют описание предметной области без ориентации на используемые в дальнейшем программные и технические средства. Однако, для построения информационной базы инфологической модели не достаточно. Необходимо провести анализ информационных потоков в системе с целью установления связи между элементами данных, их группировки в наборы входных, промежуточных и выходных элементов данных, исключения избыточных связей и элементов данных. Получаемая в результате такого анализа безызбыточная структура носит название канонической структуры информационной базы и является одной из форм представления инфологической модели предметной области.

Каноническая структура задает логически неизбыточную информационную базу. Выделение наборов элементов данных по уровням позволяет объединить множество значений конечных элементов в логические записи и тем самым упорядочить их в памяти ЭВМ.

От канонической структуры переходят к логической структуре информационной базы, а затем - к физической организации информационных массивов. Каноническая структура является также основой для автоматизации основных процессов предпроектного анализа предметных областей систем управления.

Процедуры хранения, актуализации и извлечения данных непосредственно связаны с базами данных, поэтому логический уровень этих процедур определяется моделями баз данных.

3. Базы данных.

База данных определяется как совокупность взаимосвязанных данных, характеризующихся: возможностью использования для большого количества приложений; возможностью быстрого получения и модификации необходимой информации; минимальной избыточностью информации; независимостью от прикладных программ; общим управляемым способом поиска.

Возможность использования баз данных для многих прикладных программ пользователя упрощает реализацию комплексных запросов, снижает избыточность хранимых данных и повышает эффективность использования информационной технологии. Минимальная избыточность и возможность быстрой модификации позволяет поддерживать данные на одинаковом уровне актуальности. Независимость данных и использующих их программ является основным свойством баз данных. Независимость данных подразумевает, что изменение данных не приводит к изменению прикладных программ и наоборот.

Модели баз данных базируются на современном подходе к обработке информации, состоящем в том, что структуры данных обладают относительной устойчивостью. Действительно, типы объектов предприятия, для управления которым создается информационная технология, если и изменяются во времени, то достаточно редко, а это приводит к тому, что и структура данных, обрабатываемых эти объекты, достаточно стабильна. Поэтому возможно построение информационной базы с постоянной структурой и изменяемыми значениями данных.

Каноническая структура информационной базы, отображающая в структурированном виде информационную модель предметной области, позволяет сформировать логические записи, их элементы и взаимосвязи между ними. Взаимосвязи могут быть типизированы по следующим основным видам: „один к одному“, когда одна запись может быть связана только с одной записью; „один ко многим“, когда одна запись взаимосвязана со многими другими; „многие ко многим“, когда одна и та же запись может входить в отношения со многими другими записями в различных вариантах. Применение того или иного вида взаимосвязей определило три основных модели баз данных: иерархической, сетевой реляционной.

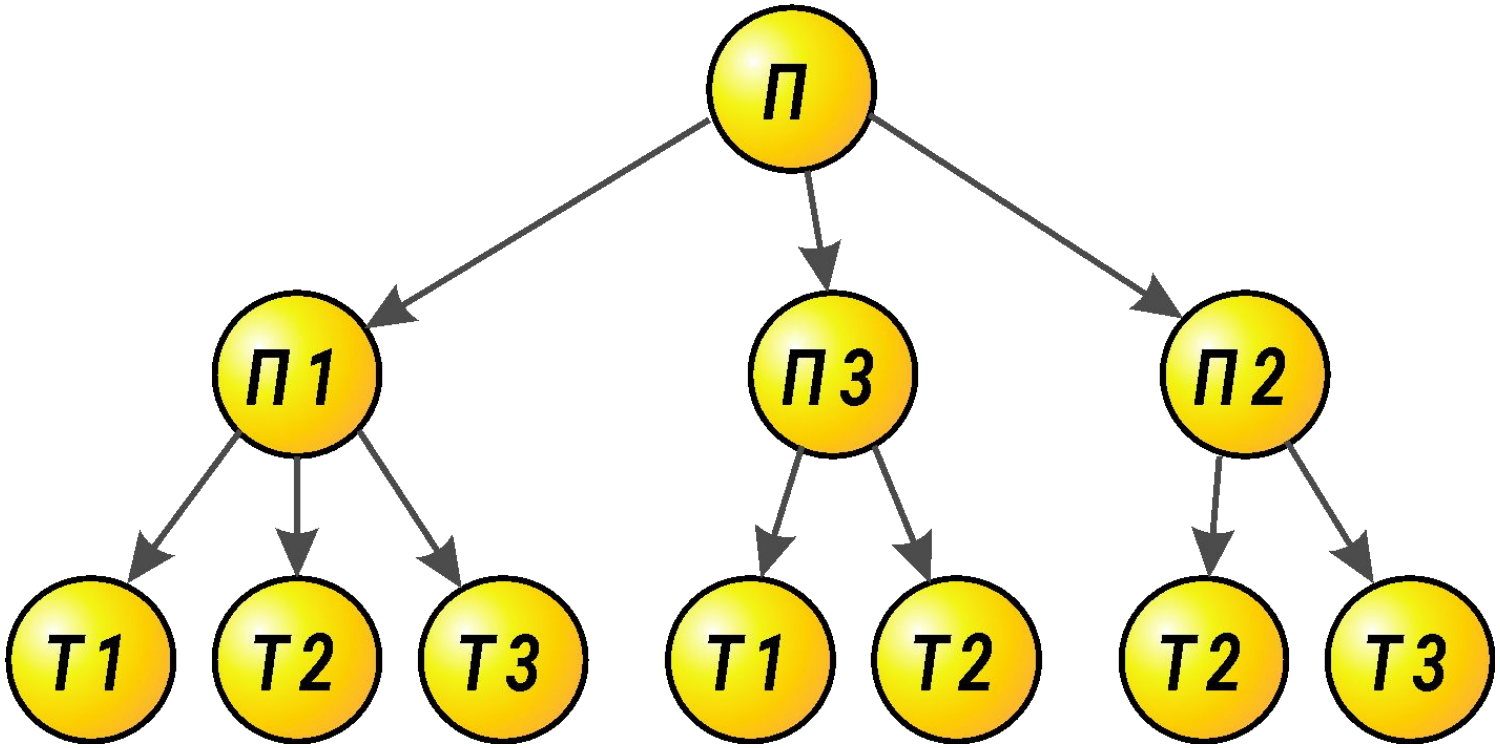
Для пояснения логической структуры основных моделей баз данных рассмотрим простую задачу: необходимо разработать логическую структуру базы данных (БД) для хранения данных о трёх поставщиках - П1, П2, П3, которые могут поставлять товары - Т1, Т2, Т3 в следующих комбинациях:

Поставщик П1 все три вида товара (Т1,Т2,Т3).

Поставщик П2 – товары Т1 и Т3.

Поставщик Т3 – товары Т2 и Т3.

Сначала построим логическую модель БД, основанную на иерархическом подходе. Иерархическая модель представляется в виде древовидного графа, в котором объекты выделяются по уровням соподчиненности (иерархии) объектов (рис. 3).



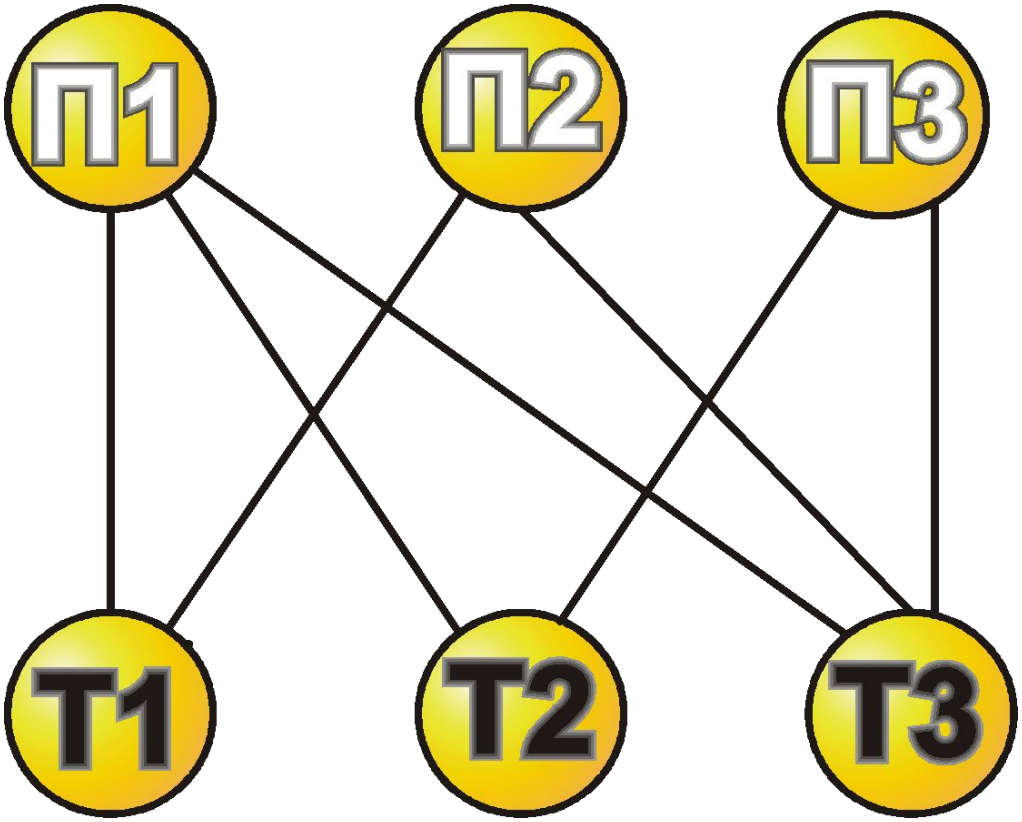
На верхнем первом уровне находится информация об объекте „поставщики“ (П), на втором - о конкретных поставщиках П1, П2 и П3, на нижнем третьем уровне - о товарах, которые могут поставлять конкретные поставщики. В иерархической модели должно соблюдаться правило: каждый порожденный узел не может иметь больше одного порождающего узла (только одна входящая стрелка); в структуре может быть только один не порожденный узел (без входящей стрелки) - корень. Узлы, не имеющие входных стрелок, носят название листьев. Узел интегрируется как запись. Для поиска необходимой записи нужно двигаться от корня к листьям, т.е. сверху вниз, что значительно упрощает доступ. Иерархическая модель данных позволяет описать их структуру как на логическом, так и на физическом уровнях. Однако, из-за жесткой фиксированности взаимосвязей между элементами данных, любые изменения связей требуют изменение структуры.

Принципиальным недостатком иерархической структуры является также жесткая зависимость физической и логической организации данных.

Быстрота доступа в иерархической модели достигнута за счет потери информационной гибкости (за один проход по дереву невозможно, например, получить информацию о том, какие поставщики поставляют, скажем, товар Т1). Указанные недостатки ограничивают применение иерархической структуры.

В иерархической модели используется вид связи между элементами данных „один ко многим“. Если применяется взаимосвязь вида „многие ко многим“, то приходят к сетевой модели данных.

Сетевая модель базы данных для поставленной задачи представлена в виде диаграммы связей на рис. 4.



На диаграмме указаны независимые (основные) типы данных П1, П2 и П3, т.е. информация о поставщиках, и зависимые - информация о товарах Т1, Т2 и Т3. В сетевой модели допустимы любые виды связей между записями и отсутствует ограничение на число обратных связей. Но должно соблюдаться одно правило: связь включает основную и зависимую запись.

Сетевая модель БД, хотя и обладает большей информационной гибкостью, но, как и иерархическая, является достаточно жесткой структурой, что препятствует развитию информационной базы системы управления. При необходимости частой реорганизации информационной базы (например, при использовании настраиваемых базовых информационных технологий) применяют наиболее совершенную модель БД - реляционную, в которой отсутствуют различная между объектами и взаимосвязями.

Реляционная модель БД обладает следующими преимуществами: простотой логической модели (таблицы привычны для представления информации); гибкостью системы защиты (для каждого отношения может быть задана правомерность доступа); независимостью данных; возможностью построения простого языка манипулирования данными с помощью математически строгой теории реляционной алгебры (алгебры отношений). Собственно, наличие строгого математического аппарата для реляционной модели баз данных и обусловило её наибольшее распространение и перспективность в современных информационных технологиях.

Для приведенной выше задачи о поставщиках и товарах, логическая структура реляционной БД будет содержать три таблицы (отношения): R1 и R2, состоящие из записей о поставщиках и о товарах соответственно, и R3 - о поставках товаров поставщиками (рис. 5).

Учитывая широкое применение реляционных моделей баз данных в информационных технологиях (особенно экономических), дадим более подробное описание этой структуры.

R1 (поставщики)

П1
П2
П3

R2 (товары)

Т1
Т2
Т3

R3 (поставка товаров)

П1	Т1
П1	Т2
П1	Т3
П2	Т1
П2	Т3
П3	Т2
П3	Т3

Рис *5.* Реляционная модель БД

РЕЛЯЦИОННАЯ МОДЕЛЬ БАЗ ДАННЫХ

Реляционная база данных это такая база данных, которая воспринимается её пользователем как совокупность таблиц. Если детализировать записи приведенного на рисунке 5 примера, то получим структуру БД, изображенную на рис. 6.

Эта база данных состоит из трех таблиц: R1, R2, R3. Таблица R1 представляет поставщиков. Каждый поставщик имеет номер, уникальный для этого поставщика, фамилию (естественно не уникальную), значение рейтинга и местонахождение (город).

Таблица R2 представляет виды товаров. Каждый товар имеет уникальный номер, название, вес и цвет.
В таблице R3 представлена поставка товаров. Она служит для того, чтобы в определенном смысле связать между собой две другие таблицы. Например, первая строка этой таблицы связывает определенного поставщика из таблицы R1 (поставщика П1) с определенным товаром из таблицы R2 (с товаром Т1). Иными словами, она представляет поставку товаров вида Т1 поставщиком по фамилии П1 и объем поставки, равный 300 штук. Таким образом, для каждой поставки имеется номер поставщика, номер товара и количество товара.

Рис.6. Реляционная БД поставщиков и товаров

R1 (поставщики)

Номер поставщика	Фамилия	Рейтинг	Город
П1	Иванов	20	Москва
П2	Петров	10	Курск
П3	Сидоров	30	Краснодар

№ товара	Название	Вес	Цвет
Т1	Гайка	12	Красный
Т2	Болт	17	Зелёный
Т3	Шайба	5	Голубой

R2 (товары)

Номер поставщика	Номер детали	Количество
П1	Т1	300
П1	Т2	200
П1	Т3	400
П2	Т1	300
П2	Т3	400
П3	Т2	200
П3	Т3	300

R3 (поставка товаров)

Из приведенных на рисунке 6 таблиц следует:
а) все значения данных являются *атомарными*, т.е. в каждой таблице на пересечении строки и столбца всегда имеется в точности одно значение данных и никогда не бывает множества значений;
б) полное информационное содержание базы данных представляется в виде *явных значений данных*. Такой метод представления - единственный, имеющийся в распоряжении реляционной базы данных. В частности, не существует каких-либо связей и указателей, соединяющих одну таблицу с другой. Для этой цели служат тоже таблицы. Так таблица R3 представляет связь таблиц R1 и R2.

Как указывалось, математическим термином для обозначения таблицы является „отношение“ (relation) и реляционные системы берут свое начало в математической теории отношений. Основы реляционной модели данных были первоначально сформулированы доктором Э.Ф. Коддом из фирмы IBM, и опубликованы в 1970 году. С тех пор эти идеи оказали широкое влияние на технологию баз данных во всех её аспектах, а так же и на другие области информационных технологий (например, искусственный интеллект и обработку текстов на естественных языках).

При работе с реляционными моделями используется как математическая терминология, так и терминология исторически принятая в сфере обработки данных. Для того, чтобы не возникало При работе с реляционными моделями используется как математическая терминология, так и терминология исторически принятая в сфере обработки данных. Для того, чтобы не возникало разночтений, ниже приведены основные формальные реляционные термины и соответствующие им неформальные эквиваленты.

Формальный реляционный термин	Неформальный эквивалент
Отношение	Таблица
Кортеж	Запись, строка
Атрибут	Поле, столбец

Реляционная модель БД имеет дело с тремя аспектами данных: со *структурой* данных, с *целостностью* данных и с *манипулированием* данными. Под *структурой* понимается логическая организация данных в БД, под *целостностью* данных понимают безошибочность и точность информации, хранящейся в БД, под *манипулированием* данными - действия, совершаемые над данными в БД. Эти три аспекта отражают и основные процедуры процесса накопления данных (хранение, актуализацию и извлечение).

Реляционная структура данных

Наименьшей единицей данных в реляционной модели является отдельное значение данных. Такие значения рассматриваются как атомарные, т.е. неразложимые, когда речь идет о данной модели. *Доменом* называют множество подобных значений одного и того же типа. Например, домен номеров поставщиков - это множество допустимых номеров поставщиков, домен объемов поставки - множество целых, больших нуля и меньших, например, 10000.

Домены представляют собой *пулы значений*, из которых берутся фактические значения, появляющиеся в атрибутах (столбцах). Смысл доменов заключается в следующем. Если значения двух атрибутов берутся из одного домена, то имеют смысл их сравнения, а, следовательно, и соединения, объединения и т.д. Если же значения атрибутов берутся из разных доменов, то всякие их сравнения лишены смысла.

Теперь определим главный элемент реляционной структуры - отношение.

Отношение на доменах *D1,D2,…,Dn*. состоит из *заголовка* и *тела*.

Заголовок состоит из такого фиксированного множества *атрибутов A1,A2,…,An*, что существует взаимно однозначное соответствие между этими атрибутами A*i* и определяющими их доменами .

Тело состоит из меняющегося во времени множества *кортежей*, где каждый кортеж в свою очередь состоит из множества пар атрибут-значений, по одной такой паре для каждого атрибута в заголовке. Для любой заданной пары атрибут-значение является значением из единственного домена , с которым связан атрибут . Если теперь посмотреть на отношения (рис. 6), то можно увидеть, что все они соответствуют приведенному определению отношения.

Строго говоря, когда мы изображаем отношение в виде таблицы, мы просто используем удобный способ представления отношения на бумаге. Таблица и отношение в действительности не одно и то же. Дело в том, что при изображении таблицы мы явно или неявно упорядочиваем расположение столбцов (атрибутов) и строк (кортежей), хотя отношение – это математическое множество, а множество в математике не обладает каким-либо упорядочением.

Значение „*n*“ - число атрибутов в отношении называется степенью отношения. Отношение степени один называется *унарным*, степени два - *бинарным*, степени три - *тернарным*, степени *n* – *n-арным*. В приведенной на рис. 6 базе данных степень отношений R1 и R2 равна четырем, а отношения R3 - пяти. Число кортежей в отношении называется *кардинальным числом* этого отношения. Кардинальные числа отношений R1, R2 и R3 на рис. равны соответственно 3, 3 и 7. Кардинальное число отношения изменяется во времени (кортеж может быть добавлен или удален) в отличие от его степени.

Целостность реляционных данных.

Важным следствием определений, сделанных выше, является то, что каждое отношение имеет *первичный ключ*, идентифицирующий это отношение. Поскольку отношение это множество, а множества по определению не содержат совпадающих элементов, никакие два кортежа отношения не могут в произвольный заданный момент времени быть дубликатами друг друга. Пусть R-отношение с атрибутами *A1, A2,...,An*. Говорят, что множество атрибутов отношения R является возможным ключом R тогда и только тогда, когда удовлетворяются два следующих независимых от времени условия:

1) уникальность;

2) минимальность.

Первое условие указывает на то, что в произвольный заданный момент времени никакие два различных кортежа отношения R не имеет одного и того же значения.

Второе условие говорит о том, что ни один из атрибутов не может быть исключен из *K* без нарушения условий уникальности.

Каждое отношение обладает, по крайней мере, одним возможным ключом, поскольку по меньшей мере комбинация всех его атрибутов удовлетворяет условиям уникальности. Один произвольно выбранный возможный ключ для данного отношения принимается за его *первичный* ключ, а остальные возможные ключи называются *альтернативными*.

Помимо первичных и альтернативных ключей, идентифицирующих данное отношение, есть еще понятие *внешнего* ключа. В общем случае внешний ключ - это атрибут или комбинация атрибутов одного отношения “R”, значение которого обязательно должно совпадать со значением первичного ключа некоторого другого отношения “R”, причем, внешний и первичный ключи должны быть определенны на одних и тех же доменах. Внешние ключи в неявном виде связывают отношения. Примером внешнего ключа является атрибут „номер поставщика“ в отношении R3 на рис. 6, поскольку этот атрибут может быть первичным ключом отношения R1. Целостность реляционной модели данных определяется двумя общими правилами.

1. Целостность по сущностям. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе базового отношения, принимал неопределенные значения. *Базовым* отношением называют независимое именованное отношение (для БД поставщиков и товаров - это отношения R1 и R2) . Мотивировка этого правила определяется тем, что базовые отношения соответствуют сущностям в реальном мире, а следовательно отличимы, т. е. имеют уникальную идентификацию. В реальной же модели функцию уникальной идентификации выполняют первичные ключи, и таким образом, ситуация, когда первичный ключ принимает неопределенное значение, является противоречивой, и говорит о том, что некоторая сущность не обладает индивидуальностью, а значит, не существует. Отсюда название „целостность по сущностям “.

2. Целостность по ссылкам.

Если базовое отношение “R” включает некоторый внешний ключ FK, соответствующий некоторому первичному ключу PK какого-либо базового отношения R', то каждое значение FK в “R” должно либо быть равным значению PK в некотором кортеже “R”, либо быть полностью неопределённым.

Неопределенность внешнего ключа может возникнуть в ситуации, когда, например, имеется вакансия на должность в некоторый отдел. Для такой должности атрибут „фамилия служащего“, являющийся внешним ключом, имеет неопределенное значение в кортеже, представляющим эту штатную должность отдела.

Манипулирование реляционными данными

Виды действий (манипуляций) над данными в реляционной модели представляют собой множество операций, получивших в совокупности название *реляционной алгебры*, и реляционной операции присваивания. Последняя производит присваивание значения некоторого производного выражения реляционной алгебры другому отношению.

Эдгар Ф. Тэд **Кодд**, **математик**, выпускник Оксфордского университета, занимаясь исследования в области ЭВМ, в 1970-м разработал понятие реляционной базы данных и определил 8 операций. Объединённых в две группы, по 4 операции в каждой.

Первая группа - *традиционные теоретико-множественные операции*.

В каждой из этих операций используется два операнда (отношения). Для всех операций, кроме декартова произведения, эти два операнда должны быть совместимы по объединению, т.е. они должны быть одной степени, и их *i*-е атрибуты () должны быть связаны с одним и тем же доменом.

Операция *объединение*.

Объединением двух отношений A и B называется множество всех кортежей *t*, принадлежащих либо A, либо B, либо им обоим. Символически эта операция показана на рис. 7. Математически эта операция записывается так:

$$A \cup B = \{t : t \in A . или . t \in B\}$$

где U-символ объединения, - знак принадлежности к определенному отношению (множеству).

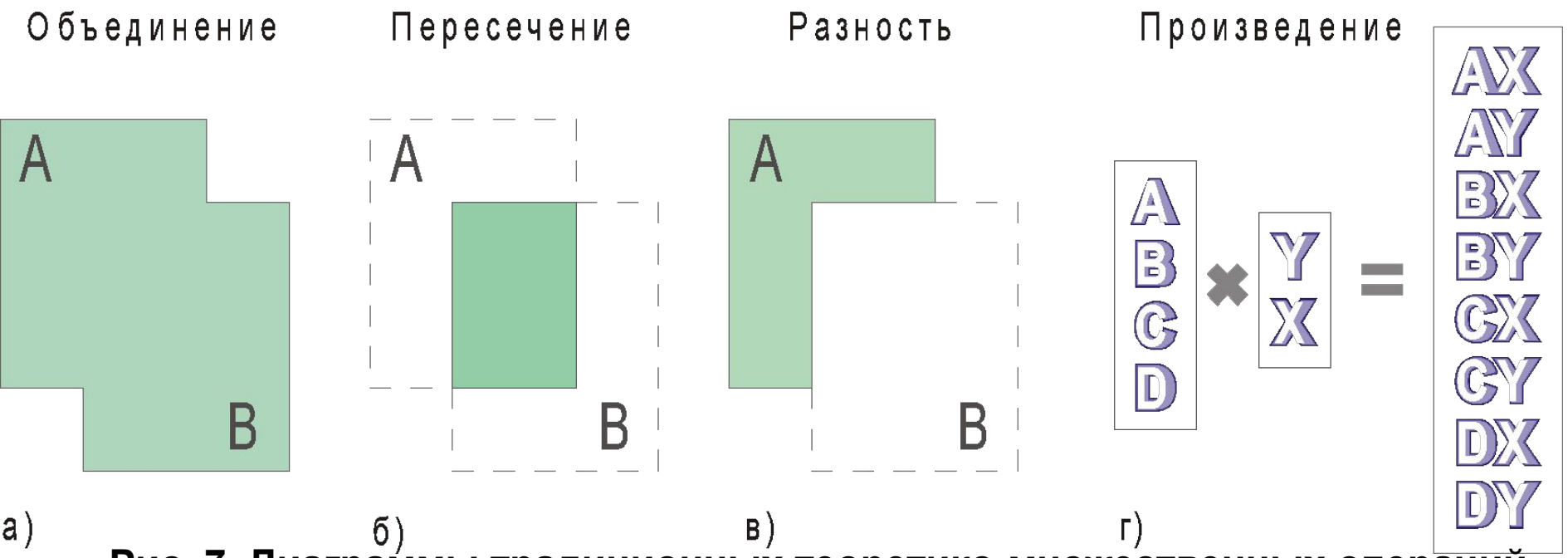


Рис. 7. Диаграммы традиционных теоретико-множественных операций

Операция *пересечения*.

Пересечением двух отношений A и B называется множество всех кортежей *t*, каждый из которых принадлежит как A, так и B (рис. 7):

$$A \cap B = \{t : t \in A \text{ и } t \in B\}$$

где ∩ - символ пересечения.

Операция *разность*.

Операция *разность*. Разностью между двумя отношениями A и B называется множество всех кортежей *t*, каждый из которых принадлежит A и не принадлежит B (рис.7):

$$A \setminus B = \{t : t \in A, t \notin B\}$$

где \ - символ разности,

$\notin$ - символ отсутствия принадлежности отношению (множеству).

Операция *декартово произведение*.

Декартовым произведением двух отношений A и B называется множество всех кортежей *t*, таких, что *t* является конкатенацией некоторого кортежа *a*, принадлежащего A, и какого-либо кортежа *b*, принадлежащего B. (конкатенация - это соединение в цепочки). Для рисунка 7 декартово произведение:

$$A \times B = \{ax, ay, bx, by, cx, cy\}$$

Вторая группа - *специальные реляционные операции*. Диаграммы этих операций приведены на рис. 8.

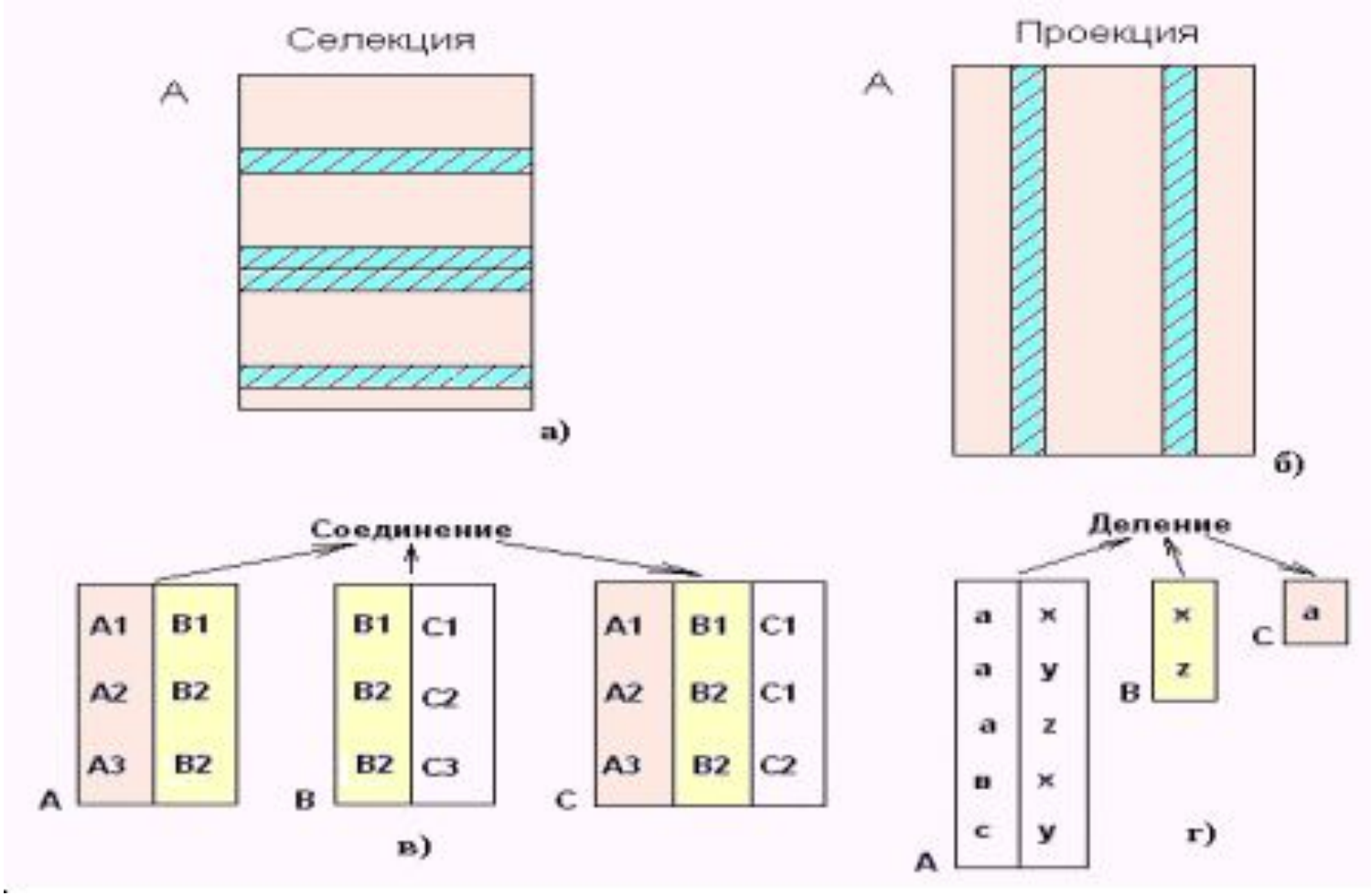


Рис. 8. Диаграммы специальных реляционных операций

Операция селекция.

Пусть *theta* представляет собой любой достижимый оператор сравнения скаляров, например, и т.п. *Theta-селекцией* отношения A по атрибутам *x* и *y* называется множество всех кортежей *t* из A, таких, что истинен предикат „*t.x theta t.y*.“ Атрибуты *x* и *y* должны быть определены на одном и том же домене, и для этого домена оператор *theta* должен иметь смысл. Вместо атрибута *y* может быть задана константа (например, выбрать из платежной ведомости записи о своих сотрудниках имеющих зарплату 500 руб.). Таким образом оператор *theta* - селекции позволит получать „горизонтальные“ подмножества заданного отношения, т.е. подмножество таких кортежей заданного отношения, для которых выполняются поставленное условие (рис. 8а).

Операция проекция

Операция проекция позволяет получить „вертикальное“ подмножество заданного отношения, т.е. такое подмножество, которое получается выбором специфицированных (определенных) атрибутов с последующим исключением, если это необходимо, избыточных дубликатов кортежей, состоящих из значений выбранных атрибутов (рис. 8б).

Операция соединение.

Пусть *theta* имеет тот же смысл, что и в операции селекции. Тогда *theta* - соединением отношения A по атрибуту *x* с отношением B по атрибуту *y* называется множество всех кортежей *t*, таких, что *t* является конкатенацией какого-либо кортежа *a*, принадлежащего A, и какого-либо кортежа *b*, принадлежащего B, и предикат „*a.x theta b.y*.“ принимает значение „истина“. При этом атрибуты A.*x* и B.*y* должны быть определены на одном и том же домене, а оператор *theta* должен иметь смысл для этого домена. Если оператор *theta* - равенство, то соединение называется *эквисоединением*. Из этого определения следует, что результат эквисоединения должен включать два идентичных атрибута. Если один из этих атрибутов исключается, что можно осуществить с помощью проекции, результат называется *естественным соединением*. Диаграмма операции приведена на рис. 8в.

Под неуточненным термином „соединение“ понимают естественное соединение. Операция „соединение“ похожа на декартово произведение. Отличие состоит в том, что декартово произведение предполагает сцепление каждого кортежа из отношения A с каждым кортежем из B, а в операции соединения кортеж из отношения A сцепляется только с теми кортежами из B, для которых выполнено условие „a.*x* = *b*.*y*.“

Операция деление.

В простейшей форме операция деления делит отношение степени два (делимое) на отношение степени один (делитель) и создает (продуцирует) результирующее отношение степени один (частное). Пусть делимое A имеет атрибуты x и y, а делитель B атрибут y(см. рис.8г). Атрибуты $A.y$ и $B.y$ должны быть определены на одном домене. Результатом деления A на B является отношение C с единственным атрибутом x, таким, что каждое значение x этого атрибута $C.x$ появляется как значение $A.x$, а пара значений (x,y) входит в A для *всех* значений y, входящих в B. Другими словами, кортеж включается в результирующее отношение C только в том случае, если его декартово произведение с отношением B содержит отношение A.

Из восьми рассмотренных реляционных операций пять являются базовыми. Это - селекция, проекция, декартово произведение, объединение и разность. Остальные три операции могут быть определены через базовые. Например, естественное соединение может быть выражено как проекция селекции декартова произведения.

Назначение реляционной операции присваивания состоит в том, чтобы сохранить значение какого-либо алгебраического выражения, например, проекции в виде заданного отношения (таблицы).

Операции реляционной модели данных представляют возможность произвольно манипулировать отношениями, позволяя обновлять БД, а также выбирать подмножества хранимых данных и представлять их в нужном виде. Таким образом, особенностями, определившими преимущества реляционной модели, являются:

а) множество объектов реляционной модели БД однородно - структура БД определяется только в терминах отношений;

б) основная единица обработки в операциях реляционной модели не запись (как в сетевых и иерархических моделях), а множество записей - отношение.

Научно-методический аппарат описания экономической информации в реляционных БД.

Все, что происходит в процессе функционирования материальных систем, может быть описано в форме сообщений. Появление сообщений о событиях, происходящих в материальной системе, представляет собой информационное отображение материальных процессов.

Сообщение может быть выражено на естественном языке, однако часто применяют форматированные сообщения, когда выделяются опорные свойства (параметры) происходящего события и в сообщении приводятся названия свойств и их значения.

Многие сообщения легко разделяются на компоненты и представляются в форматированном виде.

Форматированные сообщения - это наиболее массовый вид сообщений, хранимых и обрабатываемых в ЭИС.

База данных (БД) - это набор сообщений, которые являются истинными для соответствующей материальной системы, непротиворечивы по отношению друг к другу и к концептуальной схеме.

Сообщения в БД обычно являются форматированными и хранятся в виде единиц информации. Единицей информации называется набор символов, которому придается определенный смысл.

Минимально необходимы две единицы информации - атрибут и составная единица информации (СЕИ).

Атрибутом называется информационное отображение отдельного свойства некоторого объекта, процесса или явления. Любое сообщение записывается в форматированном виде как указание свойств (параметров) предметов, о которых мы говорим. Поэтому информационное отображение любого явления представляет собой набор соответствующим образом подобранных атрибутов.

Составная единица информации представляет собой набор из атрибутов и, возможно, других СЕИ. Простейшими СЕИ являются таблицы. СЕИ позволяет создавать произвольные комбинации из атрибутов.

Концептуальная схема (от слова concept - понятие) представляет собой описание структуры всех единиц информации, хранящихся в БД. Под структурой понимается вхождение одних единиц информации в состав других единиц информации. Следует отметить, что БД в целом также является единицей информации. Если рассматривать единицы информации как информационные объекты, то можно говорить об их свойствах. В то же время единицы информации - это нефизические объекты, так как они не занимают место в пространстве.

Простейшими характеристиками СЕИ являются имя, структура и значение. Имя СЕИ - это ее условное обозначение в процессах обработки информации. Структурой СЕИ называется вхождение одних единиц информации в состав других единиц информации.

Аппарат СЕИ рассчитан на описание структуры экономических документов. Документом называется материальный носитель информации (обычно бланк бумаги), содержащий оформленные в установленном порядке сообщения и имеющий юридическую силу.

Существует сравнительно много способов описания структуры СЕИ. Для описания, не зависимого от конкретных языков программирования и СУБД, достаточно указывать после имени СЕИ список имен входящих в нее атрибутов и СЕИ. Будем помещать этот список в круглые скобки, а имена внутри скобок перечислять через запятую. Имя СЕИ может сопровождаться размерностью, т.е. указанием на количество одинаковых по структуре значений этой СЕИ. Размерность, если она не равна 1, указывается в скобках после имени СЕИ.

Значением СЕИ называется набор значений непосредственно входящих в нее атрибутов и набор собраний непосредственно входящих в нее СЕИ. Одно значение СЕИ при хранении ее в памяти ЭВМ часто называется записью. Все языки программирования содержат средства описания структуры СЕИ.

Переименованием единицы информации называется присвоение ей нового имени, объявление синонима - это установление второго, третьего и т.д. равноценного имени для единицы информации.

Операция над значением атрибута всего одна - это перекодирование, т.е. замена существующего кода значения на новый для всех значений.

Выборка - операция выделения подмножества значений СЕИ, которые удовлетворяют заранее поставленным условиям выборки.

Корректировка означает выполнение одной из операций:

- **добавление нового значения СЕИ,**
- **исключение существующего значения СЕИ,**
- **замена некоторого значения СЕИ на новое значение.**

Декомпозиция - операция преобразования исходной СЕИ в несколько СЕИ с различными структурами. Декомпозиция, как и все операции над структурой СЕИ, одновременно производит преобразование множества значений.

Композиция - операция преобразования нескольких СЕИ с различными структурами в одну СЕИ. Декомпозиция и композиция являются взаимообратными операциями.

Нормализация - это операция перехода от СЕИ с произвольной структурой к СЕИ с двухуровневой структурой. Одновременно происходит перекомпоновка значений СЕИ.

Свертка - операция преобразования СЕИ с двухуровневой структурой в СЕИ с произвольной многоуровневой структурой.

При анализе экономических документов ставится задача разделения документа на элементарные осмысленные фрагменты, называемые показателями. Это позволяет установить смысловые взаимосвязи между различными документами, обеспечить одинаковое понимание всеми пользователями применяемых единиц информации и их единое обозначение, использовать полученные результаты для определения структуры базы данных.

Показатель представляет собой полное описание количественного параметра, характеризующего некоторый объект или процесс. Соответствующее описание произвольного свойства (необязательно количественного) называется атомарным фактом.

Чтобы точнее характеризовать атрибуты, образующие показатель, необходимо, отметить существенные различия свойств, которые отображаются атрибутами. Материальные процессы, как известно, имеют качественную характеристику и количественную характеристику. Соответственно и атрибуты должны разделяться на два класса, которые называются «атрибуты-признаки» и «атрибуты-основания». Атрибут-признак представляет собой информационное отображение качественного свойства некоторого объекта, предмета, процесса, а основание является отображением их количественного свойства. В состав показателя должны входить один атрибут-основание и несколько атрибутов-признаков, однозначно характеризующих условия существования основания.

В показателях отображаются количественные свойства объектов и процессов. Вместе с тем существуют документы, не содержащие атрибутов-оснований, например анкеты кадрового учета, сведения о структуре подразделений предприятия и т. д.

Следовательно, не вся экономическая информация может быть представлена в форме показателей.

Минимальный набор атрибутов показателя должен содержать:

- **атрибуты, отображающие идентификаторы объектов,**
- **атрибуты, отображающие признак времени,**
- **атрибут, отображающий некоторое количественное свойство объекта или взаимодействия**

Для установления признаков и оснований в конкретных документах можно использовать следующие закономерности.

1. Если значение атрибута является исходным данным или результатом арифметической операции - это основание.

2. Если значение текстовое - это признак.

3. Если атрибут обозначает предмет - это признак.

4. Если атрибут в некотором показателе является признаком (основанием), - он будет играть эту роль и в других показателях.

5. Если показатели описывают сходные процессы - их признанные части совпадают.

6. Если основание показателя вычисляется по значениям других оснований, то набор признаков такого показателя – объединение признаков, связанных с этими основаниями.

Критерием качества создания базы данных может служить минимальная избыточность хранимой информации. Обычно минимальная избыточность выражается принципом: каждое сообщение хранится в БД один раз. Соблюдение этого принципа дает ряд преимуществ:

- **сокращается объем памяти ЭВМ, требуемой для хранения базы данных,**
- **сокращается трудоемкость ввода данных в ЭВМ и упрощаются проблемы контроля достоверности вводимой информации,**
- **упрощаются алгоритмы корректировки данных, так как корректировка сообщения может быть проведена за одно обращение к базе данных.**

Использование аппарата экономических показателей позволяет создать структуру БД с минимальной избыточностью, если сначала расчленить все сведения, циркулирующие в ЭИС, на показатели, а потом объединить атрибуты родственных показателей по принципу: в один файл включается группа экономических показателей с одинаковым составом атрибутов-признаков.

Одна из причин выделения показателей в особую разновидность единиц информации заключается в том, что показатель является минимальной группой атрибутов, сохраняющей информативность (осмысленность) и поэтому достаточной для образования самостоятельного документа.

Для показателей, описывающих экономические процессы (взаимодействие объектов), можно классифицировать их составные части:

- **формальную характеристику, указывающую на алгоритм получения атрибута-основания в показателе;**
- **перечень объектов, участвующих в процессе;**
- **название процесса;**
- **единицу измерения атрибута-основания;**
- **определение момента времени или периода времени;**
- **название функции управления;**
- **название экономической системы, в которой происходит описываемый процесс.**

Показатель удобно применять как обобщающую единицу измерения объема данных.
Существует аналогия между экономическими показателями и переменными с индексами, которые рассматриваются, например, в линейной алгебре. Так, показатель П (Код материала, Цена) соответствует величине C(i), где C - цена материала с i-м Кодом материала. Переменная C соответствует атрибуту-основанию «Цена», индекс i - атрибуту-признаку «Код материала». В общем случае переменная всегда отображает атрибут-основание, а индексы этой переменной - значения соответствующих атрибутов-признаков показателя. Естественное отличие состоит в том, что индекс i переменной C обычно изменяется от 1 до некоторого фиксированного значения, а номенклатурные номера материалов (и вообще любые значения атрибутов-признаков) могут кодироваться многими способами, необязательно порядковыми кодами.

ОБЪЕКТНАЯ МОДЕЛЬ БАЗ ДАННЫХ

В последние годы все большее признание и развитие получают объектные базы данных (ОБД), толчок к появлению которых дало объектно-ориентированное программирование и использование компьютера для обработки и представления практически всех форм информации, воспринимаемых человеком.

Объектно-ориентированное программирование (ООП), в отличие от структурного, делает акцент не на программные структуры (циклы, условия и т.д.), а на объекты. Объектом называют почти все, что представляет интерес для решения поставленной задачи на компьютере. Это может быть экранное окно, кнопка в окне поле для ввода данных, пользователь программы, сама программа и т.д.

Тогда любые действия можно привязать к такому объекту, а также описать, что произойдет с объектом при выполнении опредебленных действий (например, при „нажатии“ кнопки). Многократно используемый объект можно сохранить и применять его в различных программах.

Таким образом, при ООП создают необходимые объекты и описывают действия с ними и их реакцию на действия пользователя. Если создан и определен достаточно большой набор объектов, то написание программы будет состоять в том, чтобы включить в нее и связать с собою те или иные объекты, обеспечивающие выполнение необходимых функций

Объект - **достаточно крупный блок функционально взаимосвязанных данных, при извлечении которого из ОБД включаются процедуры преобразования и отображения данных по программам, входящих в состав объекта. Типы и структуры данных, из которых состоит объект, могут быть различными у разных объектов и создаваться самим программистом на основе стандартных типов данных используемого языка программирования. Создаваемые и описываемые программистом типы данных получили название *абстрактных типов данных*.**

Таким образом, *объектом* называется программно связанный набор методов (функций) и свойств, выполняющих одну функциональную задачу. Например, кнопка управления на экране Windows - это объект, который обладает свойствами описывающими его внешний вид и назначение, и набором методов для управления его поведением на экране.

Свойство - это характеристика, с помощью которой описывается внешний вид и работа объекта.

Событие - это действие, которое связанно с объектом. Событие может быть вызвано пользователем (щелчок мышью), инициировано прикладной программой или операционной системой.

Метод - это функция или процедура, управляющая работой объекта при его реакции на событие.

Объектные модели данных еще не имеют строгой теоретической основы (как, например, реляционные), что затрудняет их создание и использование. Однако развитие средств мультимедиа, вычислительных сетей и передаче по ним аудио- и видеообъектов, заставляют интенсифицировать поиски в направлениях как создания теории, как и практической реализации надежных систем объектных баз данных.

ПРОГРАММНО – АППАРАТНЫЙ УРОВЕНЬ ПРОЦЕССА НАКОПЛЕНИЯ ДАННЫХ

Логический (модельный) уровень процесса накопления связан с физическим через программы, осуществляющие создание канонической структуры БД, схемы её хранения и работу с данными. На рис. 9 показан состав моделей процесса накопления данных и их связь с программно-аппаратным уровнем.

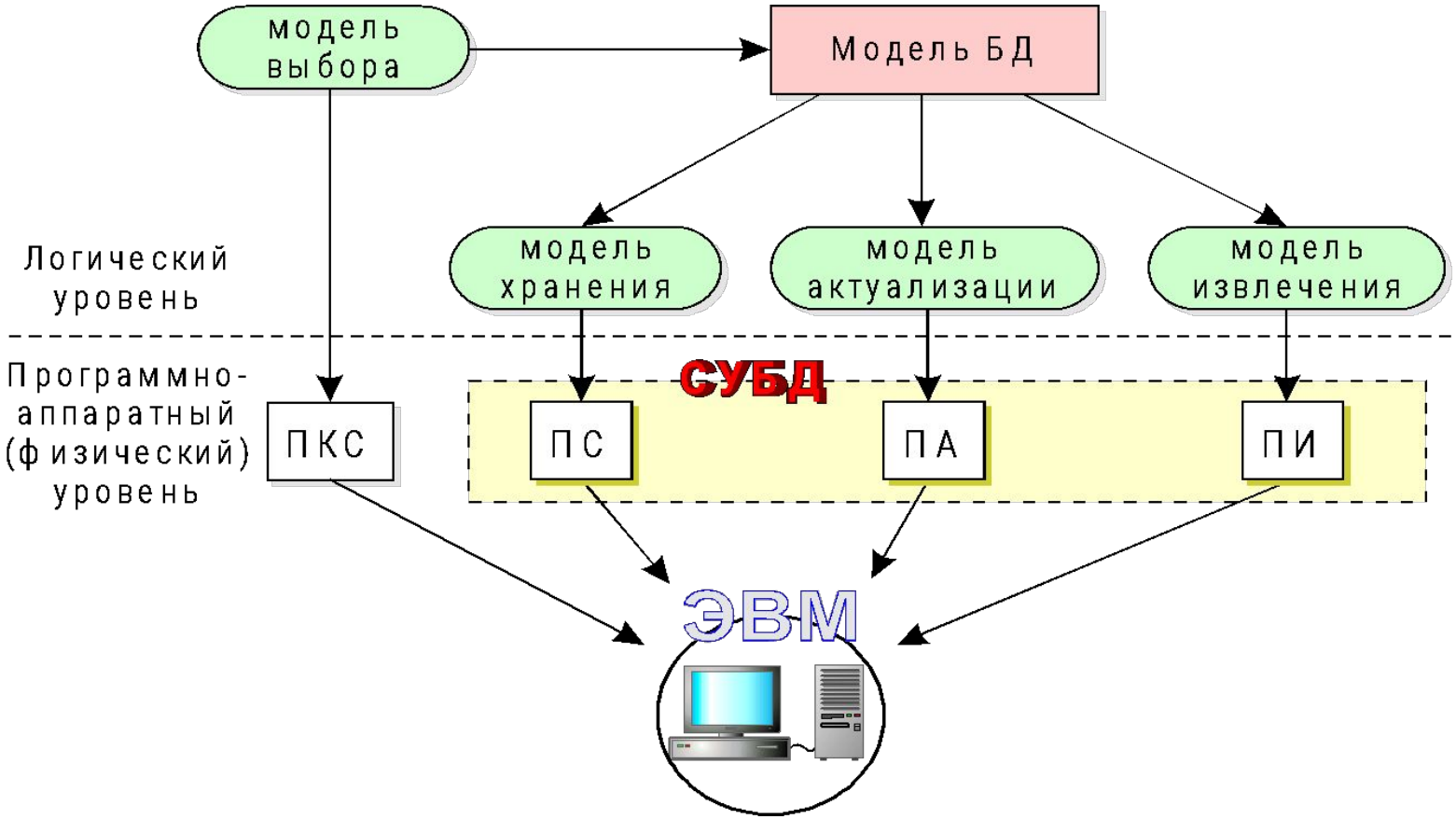


Рисунок 9. Состав моделей и программ процесса накопления данных

Каноническая структура БД создается с помощью модели выбора хранимых данных.

Формализование описания БД производится с помощью трех моделей: модели хранения данных (структура БД), модели актуализации данных и модели извлечения данных. На основе этих моделей разрабатываются соответствующие программы: создания канонической структуры БД (ПКС), создания структуры хранения БД (ПС), актуализации (ПА) и извлечения данных (ПИ)

Таким образом, переход к физической модели базы данных, реализуемой и используемой на компьютере, производится с помощью системы программ, позволяющих создать в памяти ЭВМ (на магнитных и оптических дисках) базу хранимых данных и работать с этими данными, т.е. извлекать, изменять, дополнять, уничтожать. Эти программы называются СУБД (системы управления базами данных). На рис.9 программы, входящие в СУБД заключены в пунктирный прямоугольник.

Современная СУБД содержит в своем составе программные средства создания баз данных, средства работы с данными и дополнительные, сервисные средства (рис. 10)

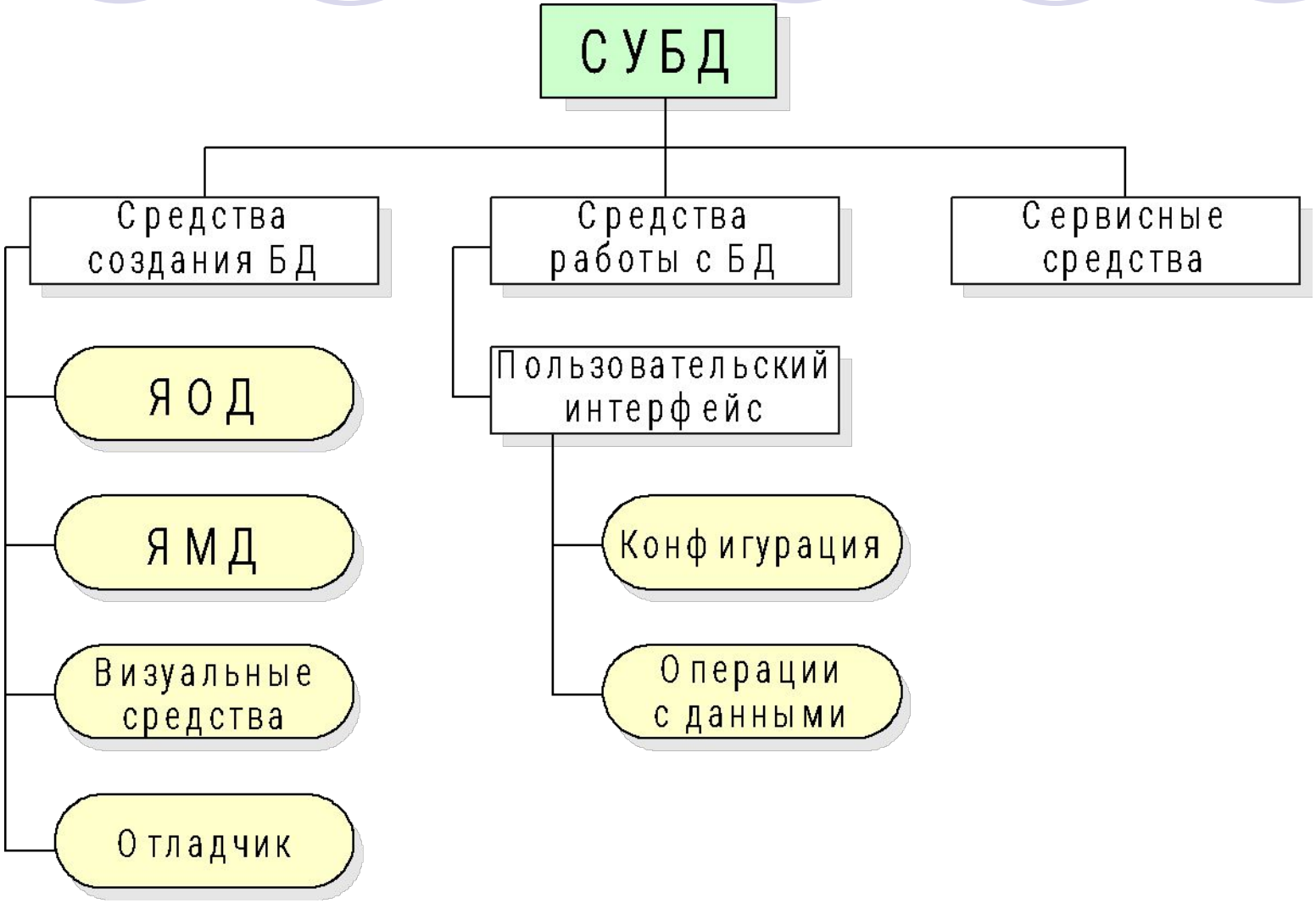


Рисунок 10. Состав системы управления базой данных.

С помощью средств создания БД проектировщик, используя язык описания данных (ЯОД), переводит логическую модель БД в физическую структуру, а на языке манипуляции данными (ЯМД) разрабатывает программы, реализующие основные операции с данными (в реляционных БД- это реляционные операции). При проектировании привлекаются визуальные средства, т.е. объекты, и программа-отладчик, с помощью которой соединяются и тестируются отдельные блоки разработанной программы управления конкретной БД.
СУБД принципиально различаются по моделям БД, с которыми они работают.
Если модель БД реляционная, то нужно использовать реляционную СУБД, если сетевая - сетевую СУБД, и т.д.

- *В технологическом, информационном процессе накопления данных наибольший вес имеют базы данных как независимые от прикладных программ хранилища данных. Однако, это не единственный способ накопления данных. Напомним, что любой вид представления информации, будь то числа, текст, программа, изображение, графический объект или звук, в ЭВМ превращается в двоичные коды - данные. Одной из форм хранения данных на дисках компьютеров, является файловая форма.* Она по-прежнему широко распространена и поддерживается всеми современными операционными системами.

- Файл - это теоретически неограниченный, статистический набор данных, физически расположенный на магнитном или оптическом диске, имеющий уникальное имя и метки начала и конца. Файлы не имеют между собой функциональной связи, но для облегчения их поиска и проведения необходимых операций, таких запись, копирование, переименование, удаление и т.п., они имеют иерархическую логическую организацию, создаваемую операционной системой компьютера. Современные операционные системы представляют пользователю разнообразный набор графических, экранных средств манипуляции файлами.