

А.И.МИШЕНИН, С.П.САЛМИН

**Теория
экономических
информационных
систем**

ПРАКТИКУМ



А.И.МИШЕНИН, С.П.САЛМИН

ТЕОРИЯ ЭКОНОМИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ, ПРАКТИКУМ

Рекомендовано
Учебно-методическим объединением
по образованию в области прикладной информатики
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальности 351400
“Прикладная информатика в экономике”
и другим междисциплинарным специальностям



Москва
“Финансы и статистика”
2005

УДК 330.47(076.5)
ББК 65с51я73
М71

РЕЦЕНЗЕНТЫ:

кафедра информационных систем в экономике
Московского университета потребительской кооперации;
И.Н. Дрогобыцкий,
доктор экономических наук, профессор

Мишенин А.И., Салмин С.П.

М71 Теория экономических информационных систем. Практикум:
Учеб. пособие. – М.: Финансы и статистика, 2005.– 192 с.: ил.
ISBN 5-279-02774-X

Практикум содержит большое количество вопросов, тестов, задач и комплексных заданий по системе понятий, описывающих экономическую информационную систему (ЭИС) в целом и ее компоненты. В практикум вошли задания по основным проблемам, моделям и методам создания и сопровождения ЭИС. Организация материала в книге связана с выделением компонентов ЭИС, этапов их жизненного цикла, уровней представления информации. Практикум методически дополняет одноименный учебник.

Для студентов вузов, обучающихся по специальности «Прикладная информатика в экономике».

М $\frac{2404000000 - 242}{010(01) - 2005}$ 378 – 2004

УДК 330.47(076.5)
ББК 65с51я73

ISBN 5-279-02774-X

© Мишенин А.И., Салмин С.П., 2005

В современных условиях возрастает значение экономических информационных систем, позволяющих обеспечить информационную поддержку процессов принятия решений в экономике.

«Теория экономических информационных систем» – один из основополагающих курсов по специальности «Прикладная информатика в экономике». Он дает студентам необходимые знания, способствующие успешному изучению дисциплин, относящихся к проектированию и организации компьютерной обработки экономических данных и внедрению автоматизированных информационных технологий в экономике и коммерции.

Основными задачами курса являются:

- изучение структуры и функций экономических информационных систем;
- выяснение роли процессов обработки данных в управлении экономическими объектами;
- применение методов и алгоритмов моделирования основных компонент экономических информационных систем;
- изучение средств и приемов информационной технологии, необходимых для приобретения основных навыков обработки данных.

Данный практикум состоит из четырех глав, комплексных заданий, тестов, контрольных вопросов и списка литературы. Практикум методически связан с учебником А.И. Мишенина «Теория экономических информационных систем» (М.: Финансы и статистика, 2003 г.), выдержавшим четыре издания (1-е и 2-е издания учебника написаны в соавторстве, 3-е и 4-е – без соавторов).

В главе 1 представлены основные компоненты экономической информации – реквизиты, показатели и документы. Закономерности построения и существования экономических показателей в документах, семантическая характеристика показателей, способы определения структуры базы данных на основе анализа показателей проиллюстрированы большим количеством заданий.

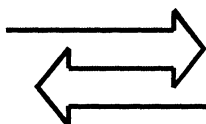
В главе 2 рассмотрена информационная алгебра для таких моделей данных, как реляционная, сетевая и иерархическая. Модели данных анализируются с учетом применяемых информационных конструкций, операций и ограничений. Подчеркиваются общие подходы, применяемые для всех указанных моделей. При разработке структуры базы данных экономических информационных систем широко применяется аппарат функциональных зависимостей. Изложены методы моделирования предметных областей экономических информационных систем, позволяющие учитывать смысловые характеристики обрабатываемой информации.

В главе 3 описаны алгоритмы и организация данных для наиболее распространенных к настоящему времени методов организации данных – последовательного, цепного и древовидного.

В главе 4 приведены комплексные задания и тесты.

Для решения задач и применения методов обработки данных предполагается использование языков семейства Foxpro и Pascal, а в качестве средств проектирования – средства Microsoft Office, в том числе Microsoft Excel и Microsoft Access.

В заданиях практикума представлены фрагменты экономических документов, заполненные условными данными.



ОСНОВНЫЕ КОМПОНЕНТЫ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ

1.1.

Элементарные свойства информации

Методические указания

Понятие «*информация*», лежащее в основе компьютерной обработки данных и информационного моделирования в любой предметной области, является фундаментальным в разнообразных процессах управления в природе, технике и обществе.

Понятие информации включает обмен ощущениями или представлениями между людьми, возможно с применением технических средств, а также отражение объективной реальности в предметах материального мира, не обязательно связанное с человеческим сознанием.

Под информацией принято понимать все, что меняет содержание знаний приемника информации относительно того предмета, о котором поступает информация. С концептуальной точки зрения в информационном процессе всегда участвуют:

- источник информации;
- материальный канал передачи информации;
- материальный носитель информации;
- приемник информации;
- предмет (содержание) информационного сообщения;
- первоначальные знания приемника информации о предмете информационного сообщения;
- приращение знаний приемника информации в результате приема информационного сообщения;
- осознание приемником информации степени полезности полученной информации.

Обмен информацией между источником и приемником показан на рис. 1.1.

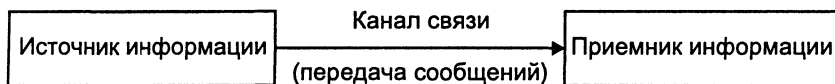


Рис. 1.1. Схема обмена сообщениями между источником и приемником информации

Любой источник и любой приемник информации содержат накопленный запас сведений (*информационную базу*) о *релевантных* (существенных для процесса и целей деятельности системы) объектах, событиях или фактах и имеют свою *предметную область* (область знаний или деятельности данного источника или приемника), а также свой *тезаурус*.

Тезаурус (греч. – сокровище) – это тематический словарь, стремящийся максимально охватить лексику передаваемых сообщений.

В информационных системах используются понятия «*релевантная информация*» и «*релевантные данные*». Свойство релевантности заключается в соответствии определенного сообщения тезаурусу определенной предметной области.

Пользователь системы компьютерной обработки данных – это специалист, профессиональная деятельность которого связана:

- либо с применением подобной системы как средства выполнения им своих функций (обычно операторских, аналитических или управленческих). Пользователи этой категории относятся к разряду конечных пользователей, обязаны владеть определенным уровнем компьютерной грамотности, но не обязаны иметь квалификацию программиста;
- либо с созданием и/или техническим обслуживанием этой системы. Данные специалисты обычно относятся к категории программистов.

Под пользователем системы обработки данных может пониматься не только отдельный специалист, но и некоторая фирма (корпоративный пользователь). Корпоративных пользователей относят к двум основным разновидностям:

- разрабатывающие фирмы – фирмы, производящие программно-информационные комплексы для массовой продажи или на заказ. Например, фирма 1С;

- фирмы, эксплуатирующие компьютерные системы. В настоящее время таковыми являются практически любые предприятия, фирмы и организации.

Классификация пользователей-специалистов более сложная и позволяет выявить различия в их функциональной и профессиональной ориентации. Рассматриваются следующие разновидности пользователей-специалистов:

- *конечный пользователь* – пользователь-непрограммист, в системах управления – сотрудник того или иного управленческого подразделения. Это основной, массовый тип пользователя компьютерных систем. Пользователи данного типа делятся на следующие подкатегории:

- а) пользователь-оператор – работник низшей или средней квалификации, выполняющий за терминалом стандартные действия (например, операционист в банке). Пользователь-оператор не имеет возможности каким-либо образом менять параметры системы, с которой работает (своего автоматизированного рабочего места – АРМ). Рабочее место пользователя-оператора обязательно оснащено «дружественным» сопряжением, в рамках которого действия пользователя строго регламентированы;

- б) профессиональный пользователь – профессионал в обслуживаемой предметной области (например, профессиональный бухгалтер), использующий автоматизированное рабочее место для выполнения своих должностных функций. Профессиональный пользователь в отличие от пользователя-оператора может быть *настройщиком АРМ*, т.е. может обращаться к макрокомандам с подстановкой параметров, как, скажем, специалист, применяющий на рабочем месте электронные таблицы собственного изготовления. Автономное *автоматизированное рабочее место (АРМ) профессионального пользователя* представляет собой человеко-машинную систему, оснащенную программно-техническим комплексом на базе персональной электронно-вычислительной машины (ЭВМ) и укомплектованную необходимыми средствами подготовки, ввода, обработки, отображения и документирования данных. Специфика работы профессионального пользователя заключается в следующем:

- цель работы – не программирование или настройка программного обеспечения, а выполнение должностных функций по своей профессиональной ориентации;

- АРМ может быть включено в локальную или глобальную компьютерную сеть;
- одно и то же оборудование может применяться несколькими пользователями или несколькими системами обработки данных.

Работники с компьютерной квалификацией (программисты и инженеры) в системе обработки экономических данных заняты либо обслуживанием, либо созданием информационных систем и автоматизированных рабочих мест. Специалисты этой категории подразделяются, в свою очередь, на следующие подкатегории:

а) прикладные программисты – специалисты, создающие прикладное программное обеспечение и использующие те или иные языки и системы программирования, или системы управления базами данных (СУБД);

б) проектировщики программно-информационных комплексов – специалисты, создающие проекты комплексных систем компьютерной обработки данных и программную реализацию подобных проектов;

в) системные инженеры, ответственные за эксплуатацию оборудования;

г) системные программисты – лица с наиболее высоким уровнем квалификации. Системные программисты в специализированных фирмах-производителях программного продукта заняты разработкой системного программного обеспечения. Однако наиболее массовая область их профессиональной деятельности – это сопровождение действующих систем в фирмах-пользователях. Системные программисты относятся к следующим профессиональным областям:

- администраторы приложений – специалисты, ответственные за штатную работу программного обеспечения;

- администраторы баз данных – специалисты, ответственные за актуальность и сохранность баз данных во внутримашинном информационном фонде;

- администраторы сети – специалисты, ответственные за правильность работы сетевого оборудования и программного обеспечения информационно-вычислительных сетей.

Пользователи и программисты любого из перечисленных классов работают за АРМ, снабженным диалоговым экранным сопряжением (интерфейсом) уровня, соответствующего их квалификации и профессиональной ориентации.

В информационных системах при передаче сообщений принято отличать понятие «информация» от понятия «данные».

Данные – это знаковые совокупности на носителе, используемые для передачи или хранения информации, расположенные, как правило, в соответствии с некоторой заданной структурой.

В сообщении на основе тезауруса воспринимаются:

1) синтаксис, или формат сообщения – совокупность правил, которым должна соответствовать структура информационных сообщений, передаваемых в виде совокупности знаков, для того, чтобы приемник мог воспринять ее и обработать;

2) семантика сообщения, т.е. смысл, то содержание, которое как раз и изменит знания приемника информации.

Данные могут нести или не нести информацию в зависимости от их соответствия тезаурусу приемника данных.

Формат (структура, синтаксис) любых данных основан на следующих понятиях:

- *алфавит* – конечная совокупность знаков (так, русский алфавит использует кириллицу, в основу английского алфавита положен латинский язык). В машинной обработке данных используется машинный алфавит, включающий кроме букв, цифры, специальные знаки, а также знаки препинания. Современные средства обработки данных обычно содержат возможность оперативного перехода от одного алфавита к другому, а также графические средства создания символов, отсутствующих в имеющихся алфавитах;

- *правила размещения данных* (собственно синтаксис, или формат). Принято различать:

а) фиксированный, или жесткий формат данных. В фиксированном формате обязательно выделены *единицы данных* (обычно небольшое количество их разновидностей), причем для каждой из единиц данных устанавливается *имя* и *область* (множество) допустимых значений, т.е. данные в фиксированном формате представляют собой совокупность значений некоторого фиксированного набора переменных (единиц данных, *реквизитов*);

б) свободный формат данных – это способ форматирования документов, при котором не выделяются специфические единицы данных, а структура сообщения строится по аналогии с сообщением на естественном языке (т.е. на том, на котором мы говорим и пишем – русский, английский и т.д.). Сообщение в свободном формате может содержать графическую и формульную информацию, рисунки и т.д.

Задание 1.1. Примените типовую схему информационного процесса (см. рис. 1.1) к следующим ситуациям:

- а) передаче телеметрической информации с искусственного спутника Земли;
- б) телефонному разговору;
- в) чтению таблицы курсов валют в газете;
- г) чтению свежих новостей в Интернете;
- д) запоминанию данных на жестком магнитном диске в конце рабочего дня, с тем, чтобы продолжить работу на следующий день;
- е) чтению древнеегипетских иероглифов в те времена, когда их расшифровка не была известна;
- ж) чтению тех же иероглифов современным специалистом-египтологом;
- з) разговору с иностранцем без переводчика;
- и) разговору с человеком, языком которого вы не владеете, через переводчика.

Задание 1.2. Рассмотрите следующие предметные области: *словесность, математика, биология, география, история*, а также перечень терминов и понятий.

Матрица. Широта. Долгота. Популяция. Склонение. Спряжение. Хребет. Тундра. Кровообращение. Баскак. Морское течение. Устье. Исток. Экватор. Тропики. Земная ось. Словарь. Перевод. Существительное. Прилагательное. Остров. Полуостров. Материк. Часть света. Румб. Возвышенность. Предложение. Расительность. Период. Низменность. Мыс. Архипелаг. Залив. Пролив. Море. Суша. Территория. Пустыня. Лесостепь. Млекопитающее. Пресмыкающееся. Обмен веществ. Нервная система. Прерия. Пунические войны. Аорта. Вена. Мышца. Скелет. Костная ткань. Империя. Алгоритм. Куликовская битва. Ритм. Алфавит. Феодализм. Принятие христианства. Арифметические действия. Квадрат. Юлий Цезарь. Квадратный корень. Полином. Логарифм. Алгоритм. Сложение. Числитель. Знаменатель. Интеграл. Бесконечный ряд. Прогрессия. Грамматика. Конъюнкция. Дизъюнкция. Синоним. Антоним. Гипербола. Ярлык. Эпитет.

Распределите понятия и термины по предметным областям. Для этого необходимо выполнить следующие действия:

- набрать самостоятельно в текстовом редакторе Word вышеприведенный перечень терминов и предметных областей;
- преобразовать текст в таблицу Термины с помощью средства Таблица/Преобразовать Панели инструментов текстового редактора Word;
- выполнить следующие операции над полученной таблицей:
 - а) создать в таблице столбцы Порядковый номер, Термин, Код термина, Предметная область, Код предметной области;
 - б) заполнить столбец Порядковый номер с помощью средства Список панели инструментов. Столбцы Предметная область и Код предметной области пока не заполняются;
 - в) присвоить терминам коды терминов в следующем порядке:
- каждой из предметных областей присваивается код из двух десятичных цифр, первая из которых не должна быть нулем. После этого Код термина строится из шести десятичных цифр, первые две из которых представляют собой Код предметной области, а четыре оставшихся – порядковый номер данного термина в данной предметной области. Скопируйте таблицу Термины в соответствующий диапазон рабочего листа табличного процессора Excel с тем же названием. На рабочем листе Excel рекомендуется:
 - пронумеровать строки таблицы с использованием средства Правка/Заполнить/Прогрессия;
 - указать названия столбцов в первой строке таблицы.

Задание 1.3. Импортируйте диапазон данных таблицы Термины табличного процессора Excel в таблицу СУБД Access, для чего необходимо вызвать на исполнение СУБД Access. Откройте в СУБД Access новую базу данных под названием Try.mdb и используйте средство Файл/Внешние данные/Импорт, которое вызовет соответствующий мастер, выполняющий необходимые действия. Создайте для последующей работы в базе данных Try.mdb, кроме таблицы Термины, также таблицу Предметные_Области, содержащую столбцы Назв_Пр_Обл и Код_Пр_Обл, которые заполняются вручную. Далее можно действовать двумя способами:

- а) в таблице Термины помещается Код_Предметной_Области без соответствующего наименования, которое будет извлекаться по связи между таблицами через поле Код_Предметной_Области (эту связь необходимо установить);
- б) в таблице Термины помещаются как Код_Предметной_Области, так и Наименование_Предметной_Области.

Задание 1.4. После построения в базе данных Tгу.mdb двух указанных таблиц дополните таблицу Термины таким образом, чтобы по каждой из предметных областей имелось не менее 10–12 терминов. Затем необходимо построить два запроса:

а) по заданному термину найти соответствующую предметную область;

б) по заданной предметной области найти все принадлежащие ей термины.

Обратите внимание, что первый из запросов может определять предметную область неоднозначно: например, термин «Гипербола» принадлежит как словесности, так и математике.

Задание 1.5. Рассмотрите приведенный ниже перечень понятий, относящихся к предметным областям *информатики* и *информационных технологий*.

Информация. Экономическая информация. Автоматизированная информационная технология. Информационно-управляющая подсистема. Управляемая (производственно-технологическая) подсистема. Управляющие воздействия. Сообщения прямой связи. Сообщения обратной связи. Запросы. Точность информации. Достоверность информации. Оперативность информации. Плановая информация. Учетная информация. Нормативно-справочная информация. Отчетно-статистическая информация. Входная информация. Выходная информация. Система обработки данных. Информационное обеспечение. Программное обеспечение. Техническое обеспечение. Правовое обеспечение. Лингвистическое обеспечение. Синтаксический анализ. Семантический анализ. Прагматический анализ. Централизованная обработка информации (данных). Децентрализованная обработка информации (данных). Абонентский пункт. Автоматизированное рабочее место (АРМ). Пакетный режим работы системы обработки данных (СОД). Интерактивный режим работы СОД. Запросный режим работы СОД. Диалоговый режим работы СОД. Учетная информация. Первичная информация. Условно-постоянная информация. Промежуточная информация. Реквизит. Имя реквизита. Область значений реквизита. Реквизит-признак. Реквизит-основание. Показатель. Экономический эффект. Экономическая эффективность. Внемашинная информационная база. Внутримашинная информационная база. Заголовочная (общая) часть документа. Предметная (содержательная) часть документа. Заверительная (оформительская) часть документа. Система класси-

фикации. Система кодирования. Классификатор. Предметная область. База данных. Файл-сервер. Клиент-сервер. Система управления базами данных (СУБД). Прикладные программы (приложения) пользователя. Системное программное обеспечение (ПО). Операционная система. Пакет прикладных программ (ППП). Алгоритм. Детерминированность. Массовость. Результативность. Дискретность. Линейная алгоритмическая структура. Ветвящийся процесс. Циклический процесс. Условие ветвления. Заголовок цикла. Тело цикла. Переменная цикла. Информационно-логическая модель предметной области. Информационный объект. Структурная связь. Формализованное представление задачи. Представление знаний. Семантическая сеть. Фрейм. Экспертная система. Тезаурус. Канал связи. Носитель. Информационная база. Релевантные данные. Источник информации. Приемник информации. Данные. Семантика. Синтаксис. Свободный формат данных. Фиксированный формат данных. Физический уровень представления данных. Хранимый уровень представления данных. Концептуальный уровень представления данных. Пользовательский (внешний) уровень представления данных. Фирма-производитель компьютеров и/или программного обеспечения (ПО). Фирма-пользователь компьютеров и ПО. Конечный пользователь. Пользователь-оператор. Профессиональный пользователь. Прикладной программист. Системный программист. Администратор базы данных БД. Администратор приложений. Администратор сети. Операция поиска. Поисковое предписание. Язык запросов. Релевантное подмножество. Индексация документов. Дескриптор. Поисковый образ документа. Поисковый шум. Поисковое молчание. Нормализованный файл. Составная единица данных (СЕИ). Запись. Нормализация документа. Модель данных. Проекция Р-файла (реляционного файла). Естественное соединение Р-файлов. Операция поиска (выборки) в Р-файле. Функциональная зависимость. Операция корректировки. Операция сортировки. Индексирование Р-файлов.

1. Выделите известные вам термины (не менее 30), остальные уберите. Полученную таблицу назовите «Термины». Составьте перечень предметных областей, к которым, по вашему мнению, относятся выбранные термины. Присвойте коды терминам и предметным областям. Составьте таблицу под названием «Предметные области», внесите в нее названия и коды предметных областей.

2. Добавьте в таблицу «Термины» справа четыре столбца. Один из них заполните пояснением значения каждого из выбранных термина, другой – кодом термина, третий – названием предметной области, четвертый – кодом предметной области. Рекомендуется формировать код термина так, чтобы он включал код предметной области.

3. Выполните над полученным таким образом материалом все действия задания 1.4. Соответствующую базу данных назовите inf.mdb.

Задание 1.6. Рассмотрите следующий перечень понятий и укажите, какие из них относятся к семантике, а какие – к синтаксису.

Алфавит значений реквизита, банковские реквизиты, длина поля, почтовые реквизиты, реквизит-основание, реквизит-признак, соответствие между почтовым кодом города и кодом Субъекта Федерации, структура документа, структура записи, юридический адрес.

Задание 1.7. Ответьте, соответствуют ли приведенные ниже сообщения тезаурусу приемника:

сообщение о нарушениях технологии, поступившее в бухгалтерию предприятия;

сообщение о прогулах, поступившее в отдел маркетинга;

зачетная ведомость, поступившая в деканат;

газета, присланная по ошибочному адресу;

жесты сотрудника ГИБДД (ГАИ), относящиеся к водителю автотранспортного средства;

прайс-лист, читаемый посетителем магазина;

прайс-лист, читаемый товароведом торгового предприятия;

бухгалтерский баланс предприятия, читаемый главным бухгалтером предприятия;

бухгалтерский баланс предприятия, читаемый аудитором;

бухгалтерский баланс предприятия, читаемый главным инженером предприятия;

учебник по любой дисциплине, изучаемой в данном учебном заведении, для учащегося этого заведения;

учебник по любой дисциплине, изучаемой в данном учебном заведении, для учащегося другого учебного заведения.

Задание 1.8. Среди следующих носителей информации укажите машинные и немашинные:

- 1) бумага;
- 2) магнитный диск;
- 3) магнитная лента;
- 4) фото пленка;
- 5) экран монитора;
- 6) компакт-диск.

Задание 1.9. Создайте таблицу «Соответствие» из трех столбцов: «Термин», «Предметная область», «Релевантность». По материалам таблиц «Термины» и «Термины из области информационных технологий» заданий 1.2–1.5 заполните два первых столбца таблицы «Соответствие» парами вида термин–предметная область. В третьем столбце укажите наличие или отсутствие релевантности для каждой из созданных подобным образом пар.

Задание 1.10. Реализуйте таблицу «Соответствие» средствами СУБД Access. Создайте запрос, позволяющий установить наличие или отсутствие свойства релевантности для заданной пары (термин – предметная область).

Задание 1.11. Приведите примеры управляющей и управляемой систем из области торговли, транспорта, производственной и финансовой деятельности.

Задание 1.12. Дайте характеристику сотрудникам согласно классификации пользователей АРМ, заполнив табл. 1.1. Дополните табл. 1.1 характеристикой типов следующих пользователей:

Главный бухгалтер (составление баланса).

Заместитель директора по кадрам (штатное расписание).

Оператор ввода товарно-транспортных накладных на складе.

Кассир в магазине с автоматизированным учетом товара.

Аудитор-аналитик, использующий в работе Excel.

Программист, использующий в работе Delphi.

Системный администратор.

Сетевой администратор.

Сотрудник, ответственный за работу прикладного программного обеспечения.

Студент-первокурсник, изучающий информатику.

Студент-дипломник, разрабатывающий собственный проект.

Выполните задание средствами Word, Excel, Access, введите и используйте коды типов пользователей.

Характеристика пользователей АРМ

Пользователь	Тип пользователя	Код типа пользователя
Операционист в банке	Профессиональный	2
...

Задание 1.13. Реализуйте средствами Access запрос к табл. 1.1 – по коду типа пользователя вывести всех пользователей этого типа.

Задание 1.14. Дайте характеристику следующих средств программного обеспечения (в том числе с точки зрения их применения в АРМ пользователями и программистами):

Табличный процессор Excel.

СУБД Access.

Бухгалтерская программа 1С-Бухгалтерия.

Программа 1С-склад.

Текстовые процессоры.

Система программирования Delphi.

Интернет-браузер Internet Explorer.

Редактор html-страниц Dream Weaver.

Задание 1.15. Сформулируйте, в чем состоит различие между информацией и данными. В каждой из позиций, приведенных в задании 1.1, укажите роль данных в информационном процессе.

Задание 1.16. Среди следующих документов укажите документы, представленные в свободном и фиксированном (жестком) формате: *деловое письмо, приказ директора, железнодорожный билет, экзаменационный билет, платежная ведомость, накладная, вексель, договор, экзаменационная ведомость, счет-фактура.*

Задание 1.17. Укажите, как называются и где применяются следующие алфавиты:

а) {А, В, С, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}.

б) {А, Б, В, Г, Д, Е, Ё, Ж, З, И, Й, К, Л, М, Н, О, П, Р, С, Т, У, Ф, Х, Ц, Ч, Ш, Щ, Ъ, Ы, Ь, Э, Ю, Я}.

- в) $\{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$.
- г) $\{0, 1\}$.
- д) $\{0, 1, 2, 3, 4, 5, 6, 7\}$.
- е) $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- ж) $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

Какое количество двоичных символов является достаточным для кодирования символов в каждом из алфавитов а) – ж)? Какие алфавиты логических значений вам известны, кроме $\{0, 1\}$?

Задание 1.18. В докладе содержится фраза: «Некоторые цеха допускают отставание от плана по ряду важнейших позиций». Содержит ли эта фраза информацию? Если да, охарактеризуйте тезаурус приемника информации. Содержит ли эта фраза данные?

1.2.

Структура экономических документов

Методические указания

Данные в экономических информационных системах могут храниться в свободном (практически по правилам естественного языка) и фиксированном форматах. В сообщениях фиксированного формата выделяются опорные характеристики (параметры) и указываются названия и значения этих параметров. Базы данных фиксированного (жесткого) формата обычно имеют табличную форму.

Базы данных свободного формата предназначены, главным образом, для визуальной диалоговой работы пользователя за терминалом (система «Консультант +» и другие подобные системы).

Запись свободного формата представляет собой машинный образ абзаца или строки документа свободного формата и обычно имеет переменную или неопределенную длину.

Файл и жесткого, и свободного форматов состоит из записей, как его элементов. Файл, как правило, содержит однородную экономическую или управленческую информацию. Разделение файла на записи обусловлено структурой алгоритма обработки данных.

Для файла свободного формата порядок записей в нем обычно существен, тогда как записи жесткого формата могут располагаться в файле в любом порядке, в зависимости от потребностей процедуры обработки. Поэтому файл записей жесткого формата можно рассматривать как множество записей, тогда как подобное утверждение для документов свободного формата в общем случае неверно.

База данных свободного формата – совокупность тематически связанных файлов в свободном формате. Отметим, что базы данных могут содержать файлы как в одном, так и в другом формате. Следовательно, внутримашинная информационная база представляет собой совокупность разнородных по формату баз данных.

Положительным качеством баз данных в свободном формате является возможность оперативного визуального обновления данных, предназначенных для некомпьютеризованного использования. Подобного рода информация применяется юристами, а также работниками налоговых служб, финансистами, бухгалтерами и др. В таких базах данных информация – это различные инструкции, законодательство, поправки к нему и пр.

Базы данных жесткого формата обычно имеют нормализованный вид, при котором реквизиты не разделяются на поименованные составные части и не образуют поименованные группы. Такая форма нормализации называется первой нормальной формой.

Основная задача проектирования базы данных экономических информационных систем (ЭИС) – определение количества файлов и их реквизитного состава.

Задача группировки реквизитов в файлы, набор которых заранее не фиксирован, допускает множество различных вариантов решений. Рациональные варианты группировки соответствуют следующим требованиям:

- множество файлов должно обеспечивать минимальную избыточность представления информации;
- корректировка файлов не должна приводить к двусмысленности или потере информации;
- перестройка набора файлов при добавлении в базу данных новых реквизитов должна быть минимальной.

Нормализация представляет собой один из наиболее изученных способов преобразования отношений, позволяющих улучшить характеристики базы данных по перечисленным критериям.

Традиционно в качестве документа рассматривают немашинный документ, внутримашинные данные рассматриваются как *об-разы* документов.

Для обработки данных важно знать тип и длину реквизита. Тип реквизита зависит от значения реквизита и от алфавита, используемого для задания его значений. Тип может быть текстовым, числовым, датой, логическим типом и т.д. Текстовый тип делится на алфавитный, цифровой и алфавитно-цифровой под-типы. Большинство кодов, применяемых на практике, относится к текстовому типу.

Длина реквизита – это максимальное количество символов, отводимое под значения данного реквизита.

Например, реквизит «сумма к выдаче», входящий в ведомость зарплаты, – это числовой реквизит (реквизит-основание). Следует отметить, что денежные реквизиты-основания имеют обычно денежный тип с двумя знаками после запятой (рубли и копейки).

Совокупность реквизитов, объединенных в более крупную единицу данных, называется составной единицей информации (СЕИ). Так, в частности, примером СЕИ может служить данное «дата», состоящее из реквизитов «год», «месяц», «число». Рассмотрим хорошо известный в практике высшей школы документ – экзаменационную ведомость. Первая страница документа изображена на рис. 1.2.

На рис. 1.3 воспроизводится часть второй страницы ведомости. В методических целях документ разделен на части, но такое деление в реальном документе отсутствует. Метки полужирным курсивом, также фактически отсутствующие в реальном документе, предназначены для анализа структуры данного документа. Значения реквизитов, которыми заполняется бланк документа, выделены светлым курсивом.

Для деления документа на части в составе большинства немашинных документов в фиксированном формате принято выделять:

1н НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.И. ЛОБАЧЕВСКОГО	
2р	ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ № <u>14</u> Группа № <u>19</u> Факультет <u>экономический (7)</u> Отделение <u>дневное</u> Специальность <u>0611</u> Дисциплина _____ <u>Общая теория систем --- (ОТС)</u> Семестр <u>2</u> Экзаменатор <u>Белов В.А.</u> Дата " <u>18</u> " <u>июня</u> 20 <u>01</u> г.
	3н ИТОГИ ЭКЗАМЕНА
	4р Отлично <u>3</u> Хорошо <u>4</u> Удовлетворительно <u>7</u> Неудовлетворительно <u>2</u> Не аттестовано <u>3</u> <div style="text-align: right; margin-top: 5px;"> Ведомость проверена <u>" 19 " июня 2001 г.</u> </div>
	5з Декан факультета _____ <div style="text-align: right;">(подпись)</div>
<p>6св ВНИМАНИЮ ПРЕПОДАВАТЕЛЕЙ:</p> <p>Перед началом экзамена студент представляет экзаменатору зачетную книжку (а при необходимости и документ, удостоверяющий личность), номер зачетной книжки заносится в экзаменационную ведомость.</p> <p>Успеваемость студентов оценивается следующими оценками: с1 – "зачтено", с0 – "не зачтено", т5 – "отлично", т4 – "хорошо", т3 – "удовлетворительно", т2 – "неудовлетворительно". Оценки "не зачтено" и "неудовлетворительно" проставляются только в экзаменационные ведомости.</p> <p>По окончании экзамена в группе экзаменатор в ведомости указывает т0 – "не явился" против фамилий студентов, отсутствовавших на экзамене (без указания причин) и расписывается, а также подводит итоги экзамена.</p>	

Рис. 1.2. Бланк экзаменационной ведомости. Первая страница

№ п/п	Фамилия и инициалы студента	Сдача зачета	Под- пись пре- пода- вателя	Экземе- национ- ная оцен- ка	Номер зачетной книжки	Подпись препо- давате- ля
8р	9р	10р	11з	12р	13р	14з
1	Ардов А. А.	зачтено		хорошо	951190	
2	Батова Т.И.	зачтено		отлично	951191	
3	Берс А.Е.	зачтено		хорошо	951192	

15з

ДЕКАН ФАКУЛЬТЕТА _____

Рис. 1.3. Бланк экзаменационной ведомости. Вторая страница

1. Общую часть документа (иногда называется заголовочной, хотя по месту расположения реквизиты общей части могут располагаться и в так называемом подножии (обычно итоговые реквизиты)).

2. Предметные строки документа – множество однородных строк данных, в большинстве случаев содержащее переменное количество подобных строк.

3. Оформительскую, или заверительную часть документа – подписи, печати и штампы, удостоверяющие юридическую силу документа и ответственность за его оформление и выполнение; оформительские реквизиты не предназначены, как правило, для хранения в виде нормализованного файла; отметим, что существуют современные технологии электронной подписи и другие методы электронной авторизации юридической силы документа, введенного в память ЭВМ, которые относятся к технологиям защиты информации в ЭВМ.

4. Реквизиты, предназначенные для улучшения читабельности немашинного документа и не подлежащие вводу в нормализованный файл, но, возможно, хранящиеся в позиции файла специального вида, например, в поле типа МЕМО.

5. Текст в свободном формате, который может храниться в виде отдельного файла или же в поле типа МЕМО.

В дальнейшем будем обозначать с помощью введенной нумерации (1, 2, 3, 4, 5) типы отдельных частей документа.

В каждом нормализованном файле существует такой реквизит или группа реквизитов, значения которой не могут совпадать ни в какой паре записей файла. Значения реквизитов, обладающих подобным свойством, естественно использовать в качестве идентификаторов записей. Подобную конструкцию принято называть *ключом* файла.

Проведем реквизитный анализ документа, для чего проанализируем формат отдельных частей документа «Экзаменационная ведомость». В составе этого документа мы видим СЕИ заголовочной части **2p** (содержащую реквизиты общего характера, тип 1, относящиеся ко всему документу, «№ экзаменационной ведомости», «№ группы», «факультет», «отделение», «специальность», «дисциплина», «семестр», «экзаменатор», «день», «месяц», «год») и **4p** (содержащую реквизиты схожего типа – «количество отличных оценок», «количество хороших оценок», «количество удовлетворительных оценок», «количество неудовлетворительных оценок», «количество неаттестованных студентов», а также СЕИ «ведомость проверена» с реквизитами «день», «месяц», «год»). Предметные строки (тип 2) в количестве, равном количеству студентов в группе, содержат для каждого из них реквизиты **8p** – порядковый номер, **9p** – фамилия и инициалы студента, **10p** – сдача зачета, **12p** – экзаменационная оценка, **13p** – номер зачетной книжки. Оформительские (заверительные) реквизиты (тип 3) имеются как в общей части – подпись декана **5з** (в заголовке) и **15з** (в подножии), так и в предметной части – **11з** и **14з** – подпись преподавателя. Константы типа 4, обозначенные **1н** и **3н**, находятся в заголовочной части документа. Наконец, часть первой страницы документа является текстом в свободном формате **бсв**.

Необходимо отметить, что текст **бсв** содержит описание области значений (перечень допустимых значений) для реквизита «сдача зачета»: {**с1** – «зачтено», **с0** – «не зачтено», **т0** – «не явился»}, а также область допустимых значений реквизита «экзаменационная оценка»: {**т5** – «отлично», **т4** – «хорошо», **т3** – «удовлетворительно», **т2** – «неудовлетворительно», **т0** – «не явился»}.

Особую роль играет часть документа, обозначенная **7сх**. Нетрудно видеть, что **7сх** – не что иное, как *схема* предметной части документа вместе с множеством значений.

Проведем нормализацию файла «Экзаменационная ведомость» путем, как уже указывалось, внесения реквизитов общей

части в строки предметной части. В нормализованном варианте используются присвоенные реквизитам машинные имена и условные значения соответствующих реквизитов. Рассмотрим результаты реквизитного анализа и нормализации документа «Экзаменационная ведомость».

Результаты анализа сведены в табл. 1.2.

Таблица 1.2

**Результат реквизитного анализа документа
«Экзаменационная ведомость»**

№ п/п	Часть документа	Тип части	Имя реквизита во внешнемашином документе	Перенос в нормализованный файл	Имя в нормализованном файле	Тип в нормализованном файле
1	<i>1н</i>	4	–	Нет	–	–
2	<i>2р</i>	1	Номер экзаменационной ведомости	Да	<i>P1</i>	Признак
3	<i>2р</i>	1	Номер группы	Да	<i>P2</i>	Признак
4	<i>2р</i>	1	Факультет	Да	<i>P3</i>	Признак
5	<i>2р</i>	1	Отделение	Да	<i>P4</i>	Признак
6	<i>2р</i>	1	Специальность	Да	<i>P5</i>	Признак
7	<i>2р</i>	1	Дисциплина	Да	<i>P6</i>	Признак
8	<i>2р</i>	1	Семестр	Да	<i>P7</i>	Признак
9	<i>2р</i>	1	Экзаменатор	Да	<i>P8</i>	Признак
10	<i>2р</i>	1	День	Да	<i>P9</i>	Признак
11	<i>2р</i>	1	Месяц	Да	<i>P10</i>	Признак
12	<i>2р</i>	1	Год	Да	<i>P11</i>	Признак
13	<i>3н</i>	4	–	Нет	–	–
14	<i>4р</i>	1	Количество отличных оценок	–	<i>Q12</i>	Основание
15	<i>4р</i>	1	Количество хороших оценок	–	<i>Q13</i>	Основание
16	<i>4р</i>	1	Количество удовлетворительных оценок	–	<i>Q14</i>	Основание
17	<i>4р</i>	1	Количество неудовлетворительных оценок	–	<i>Q15</i>	Основание

№ п/п	Часть документа	Тип части	Имя реквизита во внемашином документе	Перенос в нормализованный файл	Имя в нормализованном файле	Тип в нормализованном файле
18	4p	1	Количество неаттестованных студентов	–	Q16	Основание
19	4p	1	День	–	P17	Признак
20	4p	1	Месяц	–	P18	Признак
21	4p	1	Год	–	P19	Признак
22	5з	3	Подпись декана	Нет	–	–
23	11з	3	Подпись преподавателя	Нет	–	–
24	14з	3	Подпись преподавателя	Нет	–	–
25	15з	3	Подпись декана	Нет	–	–
26	6св	5	ВНИМАНИЮ ПРЕПОДАВАТЕЛЕЙ	Нет	–	–
27	8p	2	Порядковый номер	Нет	–	Признак
28	9p	2	Фамилия и инициалы студента	Да	P20	Признак
29	10p	2	Сдача зачета	Да	P21	Признак
30	12p	2	Экзаменационная оценка	Да	P22	Признак
31	13p	2	Номер зачетной книжки	Да	P23	Признак

Каждая из строк табл. 1.2 описывает какую-либо из частей внемашиного документа и определяет, переносится или не переносится соответствующая единица в машинный файл. Рассмотрим основные особенности проведенного анализа.

Реквизиты 1 и 13, а также 22–25 не переносятся в машинный файл, поскольку первые два являются константами, а последние четыре относятся к заверительной части документа. Реквизиты 14–21 могли бы по содержательным соображениям быть перенесены, но не переносятся просто в силу решения проектировщика, который предполагает хранить эти данные в другом файле. Часть

документа 26 не подлежит хранению в файле с жестким форматом, будучи текстом в свободном формате. Наконец, реквизит 27 не переносится в машинный файл, поскольку порядковый номер нужен только во внешнем представлении, а в машинном представлении он получается автоматически.

Каждому из реквизитов, переносимых в машинное представление файла, присваивается машинное имя, причем реквизиты-признаки получают имя, начинающееся с буквы *P*, тогда как основания обозначаются идентификаторами, которые начинаются с буквы *Q*.

Таблица 1.3

Экзаменационная ведомость, приведенная к нормализованному виду после проведения реквизитного анализа (сокращенный вариант)

2	3	4	5	6	7	8	9	10	11	12	28	29	30	31
<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>	<i>P8</i>	<i>P9</i>	<i>P10</i>	<i>P11</i>	<i>P20</i>	<i>P21</i>	<i>P22</i>	<i>P23</i>
14	19	7	Дн.	0611	ОТС	2	Белов В.П.	18	06	01	Ардов А.А.	<i>c1</i>	<i>m4</i>	951190
14	19	7	Дн.	0611	ОТС	2	Белов В.П.	18	06	01	Батова Т.И.	<i>c1</i>	<i>m5</i>	951191
14	19	7	Дн.	0611	ОТС	2	Белов В.П.	18	06	01	Берс А.Е.	<i>c1</i>	<i>m4</i>	951192

В табл. 1.3 приведен нормализованный вид полученного таким образом файла. Схема файла состоит из двух первых нумерованных строк, одна из которых содержит номера реквизитов согласно табл. 1.2, во второй строке помещены идентификаторы реквизитов согласно той же таблице. Примем в качестве имени файла, приведенного в табл. 1.2, слово *examlist*. Значения реквизитов в ячейках табл. 1.2 построены по правилам, перечисленным выше, и соответствуют значениям реквизитов в исходном документе. Ключом файла *examlist* является реквизит *P23* – «номер зачетной книжки». В символах будем записывать этот факт как *P23* = KEY (*examlist*).

Задания

Задание 1.19. Сопоставьте в табл. 1.3 описание реквизитов средствами Конструктора СУБД Access, создав таблицу *exam*. Заполните ее данными из табл. 1.3.

Задание 1.20. В следующих сообщениях (1 – 9):

- а) задайте имена реквизитов, укажите их значения;
- б) укажите реквизиты-признаки и реквизиты-основания;
- в) охарактеризуйте области значений реквизитов словесно и средствами описания данных в СУБД Access;
- г) создайте нормализованный файл средствами СУБД Access;
- д) выберите один из реквизитов и предложите для него системе кодирования.

№ п/п

Сообщение

- 1 Наименование детали: болт.
Код детали: 0107386.
Дата: 12.03.2002.
Цех: 15.
Плановый выпуск в штуках: 1500.
Фактический выпуск: 1517.
- 2 Наименование населенного пункта: Сергач.
Субъект Федерации: Нижегородская область.
Статус: районный центр.
Вид населенного пункта: город.
Население: 93 тыс. чел.
- 3 Наименование ценной бумаги: вексель.
Номинал: 1000 руб.
Процент: 9,5%.
Срок погашения: 31 марта 2001 г.
Должник: АО «Северстальинвест».
- 4 Предприятие: АО «AVON».
Годовая прибыль: 8055 тыс. руб.
Уставный капитал: 78500 тыс. руб.
Год: 2000.
- 5 Название завода: «Ока».
Расположение: г. Навашино.
Специализация: Судостроение.
Количество работающих: 5140 чел.
- 6 Поезд № 34 Москва–Кемерово прибывает на ст. Нижний Новгород 10 окт. 2001 г. в 21.15.
- 7 Изделие: принтер для ПК.
Марка: EPSON.
Тип: матричный.
Модель: LQ-1500.
Скорость печати: до 20 стр./ч.

№ п/п	Сообщение
8	Тип квартиры: 3-комнатная. Площадь: 72 м ² . Этаж: 3-й. Район: Приокский. Начальная цена: 62 000 долл.
9	Сообщение Гидрометеослужбы: «Температура + 19°, ветер южный, скорость ветра 5 м/с, атмосферное давление – 750 мм рт. ст.».

Можно ли сказать, что сообщение содержит данные? Создайте перечень реквизитов с их характеристиками и областями значений. Постройте нормализованный файл средствами СУБД Access. Добавьте реквизиты, которые вы сочтете необходимыми.

Задание 1.21. Рассмотрите табл. 1.4 «Сотрудники». Реализуйте данную таблицу средствами СУБД Access. Укажите для каждого из реквизитов имя, охарактеризуйте область значений. Проведите (не менее чем двумя способами) упорядочение по реквизитам ФИО, Табельный номер, Стаж, Профессия (по двум из них по вашему выбору). Создайте таблицу «Коды» с реквизитами Профессия и Код профессии.

Таблица 1.4

Сотрудники

Табельный номер	ФИО работника	Стаж	Год рождения	Профессия
А-3218	Сергеев А.Р.	19	1967	Слесарь
А-4001	Васильев Т.П.	10	1973	Токарь
Д-4589	Хасянов А.Б.	15	1963	Оператор станков с ЧПУ
А-6749	Моисеев И.П.	19	1969	Токарь
В-3890	Дульский П.С.	18	1970	Наладчик
В-5567	Хомченко П. Е.	12	1970	Фрезеровщик
В-4780	Майоров Д.С.	14	1972	Наладчик

Замените в таблице «Сотрудники» реквизит Профессия на реквизит Код профессии и свяжите таблицы «Сотрудники» и «Коды» по реквизиту Код профессии. Укажите в обеих таблицах реквизиты, претендующие на роль ключа.

Определите типы связей между реквизитами таблиц «Сотрудники» и «Коды». Какие связи с их типами отображены в схеме данных СУБД Access?

Задание 1.22. Пусть задана составная единица информации из реквизитов: наименование вуза, наименование факультета, наименование кафедры. Укажите не менее трех возможных связей между указанными реквизитами, определите тип связей (1:1, M:1, 1:M, M:M).

Задание 1.23. Пусть заданы следующие информационные сущности:

1. Изделие (Код_изделия, Наименование, Производитель, Цена_Производителя).

2. Поставщик (Наименование_Поставщика, Код_Поставщика, Адрес_Поставщика).

3. Поставка (Код_Поставки, Код_Поставщика, Код_Изделия, Дата, Количество, Цена).

Постройте соответствующие таблицы, определите ключи, связи, типы всех реквизитов. Выполните реквизитный анализ для указанных таблиц.

Т а б л и ц а 1.5

Значения таблицы «Поезда»

НП	М	Д	БР	НВ	ПР	КВ	КМ	КП
37	Казань–Москва	28.05.04	Шитов	1	Зуева	Плацкартный	54	45
37	Казань–Москва	28.05.04	Шитов	2	Чуева	Плацкартный	54	49
37	Казань–Москва	28.05.04	Шитов	3	Горев	Купейный	36	30
37	Казань–Москва	28.05.04	Шитов	4	Лисин	Купейный	36	18
37	Казань–Москва	28.05.04	Шитов	5	Лосев	СВ	24	5
37	Казань–Москва	28.05.04	Шитов	6	Яшина	Общий	96	27

Описание реквизитов таблицы «Поезда»

Название реквизита	Условное обозначение	Тип реквизита
Номер поезда	НП	
Маршрут	М	
Дата отправления	Д	
Фамилия бригадира	БР	
Номер вагона	НВ	
Фамилия проводника	ПР	
Категория вагона	КВ	
Количество мест	КМ	
Количество проданных мест	КП	

Задание 1.24. Рассмотрите таблицу «Поезда» (см. табл. 1.5).
Описание реквизитов таблицы «Поезда» см. в табл. 1.6:

а) расставьте в третьем столбце табл. 1.6. типы реквизитов (*P* – признак, *Q* – основание);

б) для всех реквизитов укажите размер и область значений;

в) предложите коды для тех реквизитов, для которых это целесообразно;

г) создайте таблицы кодирования и основную таблицу в базе данных СУБД Access;

д) реализуйте не менее двух запросов к таблице «Поезда» по вашему усмотрению;

е) выполните реквизитный анализ таблицы «Поезда».

Задание 1.25. Проведите реквизитный анализ, нормализацию и постройте в СУБД Access машинный образ документа «Доверенность» (рис. 1.4).

Задание 1.26. Проведите реквизитный анализ, нормализацию и постройте в СУБД Access машинный образ документа «Расходная накладная» (рис. 1.5).

Организация: Стройторг.

Доверенность № СТ000001

Дата выдачи: 21.03.04.

Доверенность действительна по: 31.03.04.

Наименование потребителя и его адрес:

Наименование плательщика и его адрес:

Фамилия лица, которому выдана доверенность: Седов А. А.

Паспорт: серия 45 05 № 640640.

Кем выдан:

На получение от: АОЗТ «Лайн».

**Перечень товарно-материальных ценностей,
подлежащих получению**

№ п/п	Наименование	Единица измерения	Количество (прописью)
1	Пылесос «Омега» 1250 Вт	шт.	Девятнадцать
2	Утюг «Тефаль»	шт.	Шесть

Подпись лица, получившего доверенность _____ удостоверяю

М.П.

Руководитель предприятия _____ Главный бухгалтер _____

Рис. 1.4. К заданию 1.25

Организация: Стройторг.

Расходная накладная № РН-0000007 от 04.03.04

Покупатель: ООО Магазин «Все для дома».

Валюта: руб.

№ п/п	Наименование товара	Цена	Количе- ство	Единица измерения	Сумма
1	Вилы	722,80	10	шт.	7228,00
2	Грабли	444,80	10	шт.	4448,00
3	Мотоблок	47260,00	1	шт.	47260,00
4	Вентилятор BINATONE ALPINE	569,90	1	шт.	569,90
5	Вентилятор настольный	104,25	1	шт.	104,25
6	Кондиционер FIRMSTAR 12M	19460,00	1	шт.	19460,00
7	Пылесос «Электросила»	289,59	1	шт.	289,59
8	Комбайн кухонный BINATONE FP 67	2418,60	1	шт.	2418,60
9	Соковыжималка BINATONE JE 102	741,33	1	шт.	741,33
10	Чайник BINATONE AFJ – 1001	750,60	1	шт.	750,60

Итого: 83270,27 руб.

В том числе НДС: 13878,39 руб.

Всего наименований 10 на сумму 83270,27 руб.

Рис. 1.5. К заданию 1.26

Экономические показатели

Методические указания

Использование аппарата экономических показателей позволяет создать структуру базы данных с минимальной избыточностью, если сначала расчленить все сведения, циркулирующие в ЭИС, на показатели, а потом объединить реквизиты родственных показателей по принципу: «В один файл включается группа экономических показателей с одинаковым составом реквизитов-признаков».

Показателем является отношение с минимальным набором реквизитов, способных образовать самостоятельный документ.

Различаются реквизиты-признаки и реквизиты-основания.

Реквизит-признак представляет собой информационное отображение качественного свойства некоторого объекта, предмета, процесса, а основание является отображением их количественного свойства.

В состав показателя должны входить: один реквизит-основание и несколько реквизитов-признаков, однозначно характеризующих условия существования основания.

Для определения признаков и оснований в конкретных документах можно использовать следующие правила:

1. Если значение реквизита является исходным данным или результатом арифметической операции, то это основание.

2. Если значение текстовое, то это признак.

3. Если реквизит обозначает предмет или время, то это признак.

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

5. Если показатели описывают сходные процессы, то их призначные части совпадают.

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

При хранении экономических показателей в памяти ЭВМ один файл отводится под группу показателей с одинаковым составом реквизитов-признаков.

Таким образом, обеспечивается проектирование файловой системы на основе анализа экономических показателей в документах.

Например, рассмотрим реквизиты документа «Приходный ордер».

Дата	Кмат – код материала	Цена
Склад	Кво-док – количество по документу	Сумма
Пост – код поставщика	Кво-пр – количество принято	

Реквизитами-основаниями являются Кво-док, Кво-пр, Цена и Сумма, которые представляют количественную характеристику процесса оприходования материала на складе. Можно сделать вывод о наличии в нашем документе четырех показателей, по одному на каждое основание. Выяснение структуры каждого показателя связано с определением реквизитов-признаков для соответствующих оснований.

У основания Кво-док необходимыми признаками будут Кмат (имеется в виду код материала), Склад и Пост (склад принимает материалы от конкретного поставщика) и Дата (нужно указать время). В результате структура показателя (назовем его П1) принимает вид

П1 (Кмат, Склад, Пост, Дата, Кво-док).

При рассмотрении показателя П2 с основанием Кво-пр можно использовать правило 5 (основания Кво-док и Кво-пр описывают сходные процессы), после чего

П2 (Кмат, Склад, Пост, Дата, Кво-пр).

Для показателя П3 с основанием Цена необходимо установить, зависят ли цены материалов от предприятия-поставщика или они постоянны. Если допустить последнее, то получим

П3 (Кмат, Цена).

Сумма в показателе П4 является результатом вычисления:

$$\text{Сумма} = \text{Кво-пр} \cdot \text{Цена},$$

поэтому согласно правилу 6 признаки показателя П4 получаются в результате объединения признаков из показателей П2 и П3, т.е.

П4 (Кмат, Склад, Пост, Дата, Сумма).

Указанные показатели образуют в базе данных два файла:

F1 с реквизитами Кмат, Цена;

F2 с реквизитами Кмат, Склад, Пост, Дата, Кво-док, Кво-пр, Сумма.

Одна из причин выделения показателей в особую разновидность единиц информации заключается в том, что показатель является минимальной группой реквизитов, 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 реквизита-основания в показателе;

- перечень объектов, участвующих в процессе;

- название процесса;

- единицу измерения реквизита-основания;

- определение момента или периода времени;

- название функции управления;

- название экономической системы, в которой происходит описываемый процесс.

Указание перечисленных выше частей необходимо для точного обозначения показателя. Реквизиты-признаки показателя должны отображать в обязательном порядке лишь перечень объектов, участвующих в процессе, и период (момент) времени. Очень часто используется признак, отмечающий единицу измерения, а остальные характеристики показателя обычно указываются в его названии, а не в хранимых значениях.

Показатель удобно применять как обобщающую единицу измерения объема данных.

Имеется аналогия между экономическими показателями и переменными с индексами, которые рассматриваются, например, в линейной алгебре. Так, показатель П3(Кмат, Цена) соответствует величине $C(i)$, где $C(i)$ – цена материала с i -м кодом материала Кмат.

Переменная C соответствует реквизиту-основанию Цена, индекс i – реквизиту-признаку Кмат. В общем случае переменная всегда отображает реквизит-основание, а индексы этой переменной – значения соответствующих реквизитов-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 реквизитов-признаков) могут кодироваться многими способами, необязательно порядковыми кодами.

Остальные показатели приходного ордера соответствуют таким переменным с индексами, как

$$\begin{aligned} K(i, j, m, n) & \text{ для П1;} \\ P(i, j, m, n) & \text{ для П2;} \\ S(i, j, m, n) & \text{ для П4,} \end{aligned}$$

где j – номер склада;
 m – код поставщика;
 n – дата.

Расчетные соотношения для показателей соответствуют выражениям для переменных с индексами, например,

$$S(i, j, m, n) = P(i, j, m, n) \cdot C(i).$$

Закономерности, установленные в математике для арифметических операций над переменными с индексами, естественно трансформируются в правила арифметических действий над показателями.

1. Пусть для показателя с числовым значением x и множеством индексов X и показателя с числовым значением y и множеством индексов Y символ @ обозначает одно из четырех арифметических действий. Тогда множество индексов Z у величины $z = x@y$ равно объединению множеств X и Y . Если множества X и Y содержат общие индексы, то необходимым условием корректности вычисления z является совпадение значений таких индексов у переменных x и y .

2. Для очень распространенных операций суммирования и умножения заимствуются правила линейной алгебры. В частности:

- если суммирование проводится по двум различным индексам, каждый из которых меняется независимо от другого, то порядок суммирования безразличен;

- если пределы изменения одного индекса зависят от другого индекса суммирования, то при перемене порядка суммирования пределы изменения каждого из индексов становятся другими.

Если индексы суммирования не указаны, то суммирование проводится по всем индексам, которые под знаком суммы встречаются два раза. Индексы, по которым ведется суммирование, называются заглушенными, индексы, по которым суммирование не ведется, называются свободными.

Представление экономической информации в форме показателей не является универсальным, так как имеются значительные массивы осмысленной экономической информации, не содержащие реквизитов-оснований (например, описания структуры экономических объектов – подразделений предприятия и т.п.).

Модель арифметических вычислений в ЭИС основывается на графе взаимосвязи показателей (или файлов). В графе $G(S, U)$ множество вершин $S = \{s(i)\}$ представляет собой все показатели (файлы), хранящиеся в базе данных. Дуга $u(i, j)$ от $s(i)$ к $s(j)$ проводится в том случае, если существует расчетное соотношение для показателя $s(j)$ и в правой части этого соотношения присутствует показатель $s(i)$.

Граф взаимосвязи показателей, дополненный параметрами потоков данных и запросов, служит основой для решения следующих задач:

- разделения промежуточных показателей на хранимые и динамически вычисляемые;

- распределения файлов по узлам вычислительной сети.

Соответствующие модели должны просчитываться заново при расширении или сокращении состава решаемых экономических задач и изменении структуры вычислительной сети.

Например, рассмотрим модель вычислений для задачи формирования программы поставок готовой продукции.

Поток входной информации содержит следующие показатели:

$\text{Пр}(i, j)$ – программа поставок i -изделия j -потребителю на внутренний рынок в год;

$\text{П}(i, j, k)$ – аналогичная программа с разбивкой по кварталам ($k = 1, 2, 3, 4$);

$\text{Эг}(i,j)$ – программа поставок i -изделия j -потребителю на экспорт в год;

$\text{Э}(i,j,k)$ – аналогичная программа с разбивкой по кварталам.

Показатели Пг и П находятся в файле прикреплений на поставку продукции П, показатели Эг и Э – в файле наряд-заказов Э. В файле цен находятся цены изделий Ц(i).

Выходные показатели:

$\text{Рг}(i)$, $\text{Р}(i,k)$ – программы поставок i -изделия на год и на квартал в натуральном выражении (файл Р);

$\text{Сг}(i)$, $\text{С}(i,k)$ – программы поставок i -изделия на год и на квартал в стоимостном выражении (файл С).

Основные расчетные соотношения имеют вид:

$$\text{Рг}(i) = \sum \text{Пг}(i,j) + \sum \text{Эг}(i,j);$$

$$\text{Р}(i,k) = \sum \text{П}(i,j,k) + \sum \text{Э}(i,j,k);$$

$$\text{Сг}(i) = \text{Ц}(i) \text{Рг}(i);$$

$$\text{С}(i,k) = \text{Ц}(i) \text{Р}(i,k).$$

Граф взаимосвязи файлов для указанной задачи приведен на рис. 1.6 а. В качестве новой задачи, которая расширяет состав графа, введем задачу оперативного контроля за формированием портфеля заказов с новыми файлами R (программа производства изделий на год и квартал) и Q (соответствие программы производства портфелю заказов). Расширенный граф взаимосвязи файлов показан на рис. 1.6 б.

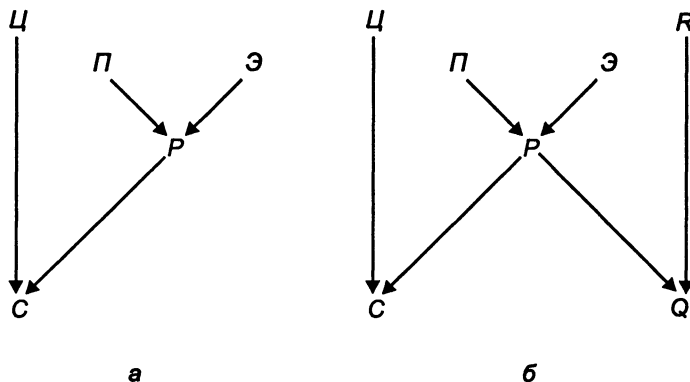


Рис. 1.6. Граф взаимосвязи файлов для задачи формирования плана поставок готовой продукции:

а – исходный граф; б – расширенный граф

Постановка экономической задачи содержит описания структуры исходных, нормативно-справочных, выходных и производных показателей, а также расчетные соотношения для вычисления выходных и промежуточных показателей, дополненные графом взаимосвязи показателей.

Можно выделить семь классов признаков, описывающих экономический процесс:

- признаки формальной характеристики;
- признаки характеристики процесса;
- признаки объекта;
- признаки единицы измерения;
- признаки субъекта;
- признаки времени;
- признаки функции управления.

Признак «формальная характеристика» отражает формальный способ вычисления соответствующего показателя. Значением данного признака могут быть следующие элементы: объем, величина, сумма, удельный вес, коэффициент, процент, прирост – средний, среднегодовой, среднесписочный и т.д.

Признак «процесс» характеризует состояние, в котором находится описываемый объект. Значением этого признака могут быть следующие элементы: наличие, выпуск, реализация, потребление.

В признаке «объект» указывается характеризующийся объектом, т.е. то, что измеряется:

- промышленно-производственный персонал, рабочие, инженерно-технические работники;
- продукция, сырье, полуфабрикаты и т.д.

Признаки «формальная характеристика», «процесс» и «объект» характеризуют общий экономический смысл показателя и являются основными.

Признаки «единица измерения», «субъект», «время» и «функция управления» позволяют конкретизировать показатель до идентификации единственного количественного значения.

К признаку «единица измерения» относятся натуральные единицы измерения (штуки, тонны, метры), трудовые единицы измерения, стоимостные единицы измерения (рубли), временные единицы измерения (год, квартал, месяц, день).

К признаку «субъект» относятся термины, обозначающие предприятия и организации, которые производят действия над

характеризуемым объектом (например, Московский машиностроительный завод).

В признаке «время» указывается конкретный момент, к которому относится показатель.

Признак «функция управления» указывает на принадлежность показателя к определенной функции управления.

На основе приведенной выше классификации признаков в экономическом показателе можно составить тезаурус для описания, например, показателей трудовых ресурсов.

*1. Множество значений признака
«формальная характеристика»,
образующих абсолютные показатели*

Наименование свойств	Код
Размер	0101
Число	0102

*2. Множество значений признака
«формальная характеристика»,
образующих относительные показатели*

Наименование свойств	Код
Удельный вес	0201
Коэффициент	0202
Процент выполнения	0203
Рост	0204
Прирост	0205
Изменение	0206
Средний	0207
Среднегодовой	0208
Среднесписочный	0209

3. Множество значений признака «процесс»

Наименование свойств	Код
Численность	0301
Количество	0302
Наличие	0303

4. Множество значений признака «объект»

Наименование свойств	Код
Промышленно-производственный персонал	0401
Ученики	0402

Рабочие	0403
Инженерно-технические работники	0404
Служащие	0405
Вспомогательные рабочие	0406
Непромышленный персонал	0407

5. Множество значений признака «единица измерения»

Наименование свойств	Код
Человек	0501

6. Множество значений признака «субъект»

Наименование свойств	Код
Отрасль машиностроения	0601
Московский машиностроительный завод	0602

7. Множество значений признака «время»

Наименование свойств	Код
2000 г.	0701
2001 г.	0702
2002 г.	0703
2003 г.	0704
I кв.	0705
II кв.	0706
III кв.	0707
IV кв.	0708
Месяц январь	0709
...	
Месяц декабрь	0720
Предыдущий год	0721

8. Множество значений признака «функция управления»

Наименование свойств	Код
Фактический	0801
Ожидаемый	0802
Нормативный	0803

Предметной областью называются элементы материальной системы, информация о которых хранится и обрабатывается в ЭИС. Для описания предметной области необходимы четыре понятия – объект, свойство объекта, взаимодействие объектов и свойство взаимодействия.

В экономических приложениях понятие «объект» сужается до понятия «физический объект», под которым понимается любой предмет, занимающий место в пространстве.

Классификация экономических объектов начинается с выделения средств производства, предметов труда и исполнителей.

Свойством объекта называется величина, характеризующая объект в каждый момент времени. Определение объекта сводится к выделению его свойств. Два объекта являются различными, если они отличаются по значению хотя бы одного свойства.

Взаимодействием объектов называется факт участия нескольких объектов в каком-либо процессе, который протекает и во времени и в пространстве.

Свойством взаимодействия называется такое свойство, которое характеризует совместное поведение объектов, но не относится ни к одному объекту в отдельности.

Задания

Задание 1.27. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Расход материалов»

Участок	Кмат – код материала	Цена
Деталь	Количество	Сумма
Операция		

Задание 1.28. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Расчет загрузки оборудования»

Цех	Количество станков	Коэффициент загрузки станка
Квартал	Код детали	
Код станка	План производства деталей	

Задание 1.29. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Ведомость сборки изделия»

Код изделия	Цена изделия	Количество деталей на узел
Код узла	Цена детали	Количество деталей на изделие
Код детали		

Задание 1.30. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Расчет численности рабочих»

Участок	Норма времени на выпуск продукции	Плановое рабочее время
Квартал	Процент выполнения норм Процент потерь времени	Количество рабочих

Задание 1.31. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Оборотная ведомость движения материалов»

Склад	Цена	Расход
Дата	Начальный остаток	Конечный остаток
Код материала	Приход	

Задание 1.32. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Ведомость отгруженных изделий»

Дата отгрузки	Цена	Сумма всего
Код изделия	Общая стоимость изделий	
Количество отгружено	Стоимость комплекта тары	

Задание 1.33. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Производственная программа участка»

Месяц	Код детали	План выпуска
Участок	План запуска	Остаток деталей
	Остаток прошлого месяца	

Задание 1.34. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Месячная программа участка»

Цех	Год	План выпуска на месяц
Участок	Код детали	План выпуска на квартал
Месяц	Расценка	

Задание 1.35. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Рапорт о выработке»

Дата	Код детали и операции	Расценка
Участок	Разряд работы	Сумма
Табельный номер рабочего	Принято деталей	

Задание 1.36. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Счет за электроэнергию»

ФИО	Дата	Сумма
Номер счета	Показание счетчика на дату	
Адрес	Расход электроэнергии	

Задание 1.37. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Нормативная калькуляция на детали»

Участок	Расход материалов (руб.)	Норматив себестоимости
Деталь	Норма времени	
Операция	Норматив зарплат	

Задание 1.38. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Выпуск деталей»

Цех	Количество станков	Фактический выпуск
Квартал	Код детали	
Код станка	План производства деталей	

Задание 1.39. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Обеспеченность материалами по цеху»

Цех	Приход материала	Отклонение
Месяц	Потребность на месяц	
Код материала	Остаток прошлого месяца	

Задание 1.40. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Акт о браке»

Дата	Операция	Себестоимость единицы брака
Участок	Табельный номер виновника	Сумма потерь от брака по участку
Деталь	Количество брака	

Задание 1.41. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Листок учета простоев»

Дата	Табельный номер рабочего	Сумма оплаты
Цех	Код причины простоя	Продолжительность простоя
	Процент оплаты простоя	

Задание 1.42. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «План поставок на год»

Год	Адрес	План поставок в руб.
Изделие	Цена изделия	
Потребитель	План поставок в шт.	

Задание 1.43. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

*Реквизиты документа «Ведомость на выдачу
заработной платы»*

Год, месяц	Табельный номер рабочего	Сумма удержания
Цех	Начислено по табельному номеру	Сумма выдано
ФИО рабочего	Сумма аванса	

Задание 1.44. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

*Реквизиты документа «Расход материалов
на изготовление изделия»*

Код изделия	Единица измерения	Количество изделий
Месяц	Норма расхода на изделие	Фактический расход материала за месяц
Код материала	Фактический расход на изделие	

Задание 1.45. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

*Реквизиты документа «Лимитная карта
на отпуск материалов»*

Дата	Цех-получатель	Цена
Код материала	Отпущено материала	Сумма
Цех-отправитель	Остаток материала	

Задание 1.46. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать само-

стоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Карточка водителя»

Табельный номер	Разряд	Часы работы
ФИО водителя	Номер путевого листа	Оплата по тарифу
Номер автомашины	Дата	Надбавка за ремонт

Задание 1.47. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Кассовый отчет кинотеатра»

Кинотеатр	Фильм	Число проданных мест
Число мест	Режиссер	Выручка от фильма
Дата		

Задание 1.48. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Нормы расхода материалов»

Цех	План выпуска деталей	Год, месяц
Код материала	Цена материала	
Код детали	Норма расхода материала на деталь	

Задание 1.49. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Учет готовой продукции предприятия»

Код предприятия	Цена продукции	Процент налога на добавленную стоимость
Код продукции	Годовой оборот предприятия	
Год	Прибыль	

Задание 1.50. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

*Реквизиты документа
«Акт о ликвидации основных средств»*

Название объекта	Код цеха	Сумма износа
Год изготовления	Дата ликвидации	Остаточная стоимость
Инвентарный номер	Первоначальная стоимость	

Задание 1.51. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

Реквизиты документа «Ведомость резки заготовок»

Код детали	Выпуск изделия за год	Цех-получатель детали
Код изделия	Количество деталей из заготовки	Норма отходов на деталь
Код заготовки		

Задание 1.52. Даны реквизиты документа. Укажите реквизиты-признаки и реквизиты-основания. Запишите структуру экономических показателей. Обозначения реквизитов выбрать самостоятельно. Сколько файлов потребуется для хранения выделенных показателей в памяти ЭВМ?

*Реквизиты документа
«Нормативная калькуляция на изделие»*

Участок	Изделие	Норматив себестоимости
Деталь	Норма времени	
Операция	Норматив зарплаты	

Задание 1.53. Составьте тезаурус для описания экономических показателей предметов труда (рис. 1.7).

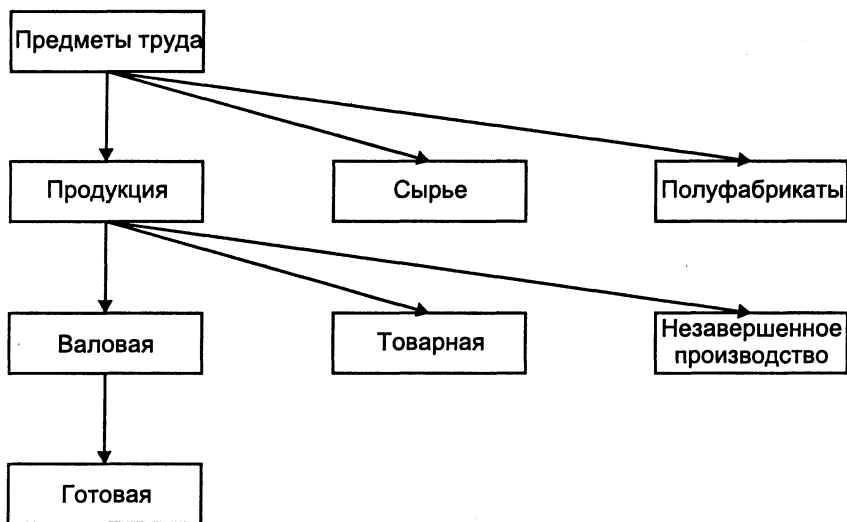


Рис. 1.7. Схема тезауруса предметов труда

Предусмотрите показатели для пяти машиностроительных заводов – московского, владимирского, ростовского, воронежского и саратовского.

Задание 1.54. Для перечисленных ниже экономических документов приведите формулы вычисления показателей и граф алгоритмической связи показателей.

Предложения к проекту программы производства продукции

Наименование изделия	Единица измерения	2001	2002		2003	В том числе по кварталам			
		отчет	план	А	проект	I	II	III	IV

А – ожидаемое выполнение.

Проект программы ремонтов

Наименование изделия	Единица измерения	2001	2002		2003
		факт	план	А	проект

А – ожидаемое выполнение.

Книга нормативов

Наименование материала	Наименование изделия	Расход на единицу изделия		
		2001	2002	2003

Задание 1.55. Для перечисленных ниже экономических документов приведите формулы вычисления показателей и граф алгоритмической связи показателей.

Программа производства цеха _____ на 20__ г.

Наименование материала	Единица измерения	План

Программа ремонта цеха _____ на 20__ г.

Наименование материала	Единица измерения	План

План распределения материала _____ на 20__ г.

Получатель	Выделено на 20__ г.

Задание 1.56. Даны реквизиты документа «Рапорт о выработке».

Реквизиты документа «Рапорт о выработке»

Дата	Код детали и операции	Расценка
Участок	Разряд работы	Сумма
Табельный номер рабочего	Принято деталей	

а) Реквизит «принято деталей» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

б) Реквизит «расценка» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

в) Реквизит «дата» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

Задание 1.57. Даны реквизиты документа «Ведомость отгруженных изделий».

Реквизиты документа «Ведомость отгруженных изделий»

Дата отгрузки	Цена	Сумма всего
Код изделия	Общая стоимость изделий	
Количество отгружено	Стоимость комплекта тары	

а) Реквизит «цена» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

б) Реквизит «код изделия» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

в) Реквизит «отгружено» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

Задание 1.58. Даны реквизиты документа «Счет за электроэнергию».

Реквизиты документа «Счет за электроэнергию»

ФИО	Дата	Сумма
Номер счета	Показание счетчика на дату	
Адрес	Расход электроэнергии	

а) Реквизит «адрес» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

б) Реквизит «ФИО» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

в) Реквизит «сумма» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.
Укажите номер правильного ответа.

Задание 1.59. Даны реквизиты документа «Нормативная калькуляция на детали».

Реквизиты документа «Нормативная калькуляция на детали»

Участок	Расход материалов (руб.)	Норматив себестоимости
Деталь	Норма времени	
Операция	Норматив зарплаты	

а) Реквизит «деталь» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

б) Реквизит «норма времени» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

в) Реквизит «участок» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

Задание 1.60. Даны реквизиты документа «Оборотная ведомость движения материалов».

Реквизиты документа «Оборотная ведомость движения материалов»

Склад	Цена	Расход
Дата	Начальный остаток	Конечный остаток
Код материала	Приход	

а) Реквизит «склад» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

б) Реквизит «дата» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

в) Реквизит «расход» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

Задание 1.61. Даны реквизиты документа «Производственная программа участка».

Реквизиты документа «Производственная программа участка»

Месяц	Код детали	План выпуска
Участок	План запуска	Остаток деталей
	Остаток прошлого месяца	

а) Реквизит «код детали» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

б) Реквизит «план выпуска» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

в) Реквизит «месяц» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

Задание 1.62. Даны реквизиты документа «Расчет загрузки оборудования».

Реквизиты документа «Расчет загрузки оборудования»

Цех	Количество станков	Коэффициент загрузки станка
Квартал	Код детали	
Код станка	План производства деталей	

а) Реквизит «цех» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

б) Реквизит «количество станков» представляет:

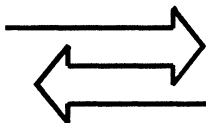
1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.

в) Реквизит «код детали» представляет:

1 – объект; 2 – свойство объекта; 3 – свойство взаимодействия.

Укажите номер правильного ответа.



ИНФОРМАЦИОННАЯ АЛГЕБРА

2.1.

Операции над отношениями

Методические указания

Модель данных – это совокупность трех составляющих:

- – множество информационных конструкций, допускаемых этой моделью;
- – множество допустимых операций над данными;
- – множество ограничений, наложенных на информационные конструкции.

Наиболее распространенные модели данных:

- реляционная;
- сетевая;
- иерархическая.

Модель данных – это инструмент для представления данных в базе данных.

В реляционной модели данных информационной конструкцией является отношение (таблица); операциями – проекция, выборка и соединение; ограничением – функциональная зависимость.

Отношение (таблица) в реляционной модели данных обладает следующими свойствами.

1. Имеет фиксированное количество столбцов и переменное количество строк. При этом строки, как правило, соответствуют отображаемым реальным событиям, фактам или объектам, а столбцы – свойствам, или характеристикам этих событий, фактов или объектов. Поэтому каждая строка характеризуется одним и тем же набором свойств.

2. На пересечении строки и столбца всегда находится единственное значение соответствующего свойства.

3. Порядок столбцов таблицы, как правило, фиксирован, но это не существенно. Каждый столбец таблицы имеет уникальное название, или имя. При этом элемент данных может иметь несколько имен, ни одно из которых не может совпадать с именем какого-либо другого элемента.

4. Любые две строки таблицы различаются, по крайней мере, одним значением некоторого столбца.

5. Порядок строк таблицы несуществен, он может меняться в процессе обработки данных.

Строки определенной подобным образом таблицы данных принято называть записями. Каждый элемент таблицы рассматривается при обработке данных как неразложимый на более мелкие элементы и называется *реквизитом*, или *атрибутом*. Каждый из столбцов таблицы (файла):

- отображает свойства, характеризующие строки таблицы;
- содержит (в клетках) значения этих свойств, относящиеся к соответствующей строке;
- имеет *имя столбца*, которое считается также именем реквизита и входит в *схему* файла, (тогда как значения, находящиеся в клетках столбца, относятся к *экземпляру* файла);
- имеет *домен*, или *область значений* величин или слов, находящихся в клетках столбца, причем значения клеток могут быть взяты только из соответствующего домена.

Клетки машинного представления таблицы называются *полями* реляционного файла, или *R-файла*.

Таблицу данных, обладающую перечисленными свойствами, принято называть *отношением*, *нормализованным файлом*, *реляционным файлом*, *реляционной базой данных* (в последнем случае речь идет о совокупности тематически связанных таблиц). При этом перечень имен реквизитов файла (с возможной дополнительной информацией о реквизитах – тип, длина и пр.) называется *схемой* файла, тогда как совокупность значений реквизитов, находящихся в строках (записях) файла, называется *экземпляром* файла. Экземпляр файла *F* обозначается $EX(F)$.

Свойство 3 нормализованного файла можно понимать в том смысле, что каждый реквизит повторяется в схеме файла не более одного раза.

Реляционный файл представляет собой машинный образ некоторого нормализованного немашинного документа фикси-

рованного формата, или же некоторой совокупности подобных документов. Каждая строка реляционного файла может соответствовать одному документу или одной строке некоторого документа. В таком, характерном для бухгалтерских систем, файле, как журнал операций, каждая запись соответствует одной бухгалтерской проводке. Проводка – это строка данных в жестком формате с реквизитами: дата, номер проводки, счет по дебету, счет по кредиту, сумма, описание.

Таблица 2.1

Файл F_0

A	B	C	D	E
$a1$	$b1$	$c1$	$d1$	$e2$
$a2$	$b2$	$c2$	$d1$	$e2$
$a3$	$b3$	$c3$	$d2$	$e5$
$a4$	$b1$	$c4$	$d3$	$e2$

Рассмотрим представление файла F_0 в табл. 2.1. В первой строке данной таблицы находится перечень имен реквизитов (структура или схема нормализованного файла) – A, B, C, D, E . Все остальное – это экземпляр файла, состоящий из четырех записей (строка – это запись). Каждая запись файла содержит в точности одно значение каждого из реквизитов. Значения реквизитов условно обозначены строчными латинскими буквами с индексами, совпадение букв (с индексами) соответствует совпадению значений реквизитов.

Набор значений в каждой из колонок является подмножеством области значений (домена) соответствующего реквизита. Перечень реквизитов реляционного файла представляет собой его схему (структуру), схема $CX(F_0)$ файла F_0 – это $CX(F_0) = \langle A, B, C, D, E \rangle$.

Файл F_0 , содержащий четыре записи (как и любой конкретный экземпляр файла), не позволяет дать заключение относительно областей значений реквизитов, однако можно утверждать, что $\{a1, a2, a3, a4\} \subseteq \text{DOM}(A)$, $\{b1, b2, b3\} \subseteq \text{DOM}(B)$, $\{c1, c2, c3, c4\} \subseteq \text{DOM}(C)$, $\{d1, d2, d3\} \subseteq \text{DOM}(D)$, $\{e2, e3, e5\} \subseteq \text{DOM}(E)$, где $\text{DOM}(A)$, $\text{DOM}(B)$, $\text{DOM}(C)$ и т.д. обозначают области значений соответствующих реквизитов.

Экземпляр $EX(F0)$ файла $F0$ удовлетворяет соотношению

$$EX(F0) \subseteq \text{DOM}(A) \times \text{DOM}(B) \times \text{DOM}(C) \times \text{DOM}(D) \times \text{DOM}(E),$$

т.е. представляет собой подмножество декартова произведения областей значений реквизитов (знак \times означает декартово произведение).

Множество, построенное таким способом, в математической терминологии принято называть отношением между элементами областей значений реквизитов; при этом реляционные операции естественно называть операциями над отношениями. Если схема содержит n реквизитов, то отношение называют n -арным.

Основными операциями над отношениями являются проекция, выборка и соединение. Все операции рассматриваются в алгебраической записи и в нотации языка SQL.

Центральным средством доступа к базе данных в языке SQL является команда Select и ее параметры Into, From, Where, Group by, Having, Order by.

В команде Select указываются имена выводимых реквизитов. Параметр From является обязательным и содержит имена требуемых для выполнения запроса отношений. Параметр Where определяет условия, которым должны удовлетворять выводимые данные. В записи условий применяются знаки сравнения ($>$, $=$ и т.д.), опции All, Any, Between, Exists, Like, In и логические операторы. Параметр Group by объединяет записи с одинаковым значением некоторого реквизита. Параметр Having при необходимости проверяет условия внутри группы записей, выделенных с помощью Group by. Параметр Order by определяет имена реквизитов, по которым должен быть отсортирован результат.

Проекция использует одно исходное отношение и создает одно результирующее отношение. В результирующее отношение переносятся те столбцы исходного отношения, которые удовлетворяют условию проекции.

Алгебраическая запись проекции имеет вид:

$$T = R[X],$$

где T – результирующее отношение;

R – исходное отношение;

X – список реквизитов в структуре отношения T (условие проекции).

Для проекции должно соблюдаться условие $CX(T) \subseteq CX(R)$.

Задания, связанные с проекцией, выполняются с помощью операции Группировка над соответствующими таблицами СУБД Access.

Для рассмотренного выше файла $F0$ проекция на столбцы A и B (обозначается $T = F0[A, B]$) имеет вид:

T	A	B
	$a1$	$b1$
	$a2$	$b2$
	$a3$	$b3$
	$a4$	$b1$

Данные столбцы как бы «вырезаются» из исходного файла. Результатом проекции является нормализованный файл, поэтому результат операции содержит равное или меньшее количество записей по сравнению с исходным файлом (в данном случае – равное).

Эти же действия на языке SQL записываются в следующем виде.

```
SELECT A, B INTO ABProF0  
FROM F0  
GROUP BY A, B;
```

Здесь:

```
SELECT A, B FROM F0 – выбор столбцов;  
INTO ABProF0 – создание новой таблицы с именем ABProF0;  
GROUP BY A, B – удаление совпадающих строк.
```

Выборка использует одно исходное отношение и создает одно результирующее отношение. В результирующее отношение переносятся те столбцы исходного отношения, которые удовлетворяют условию выборки. Условие выборки проверяется в каждой строке исходного отношения по отдельности и не может охватывать информацию из нескольких строк. Существуют две простейшие разновидности условия выборки:

1) Условие вида Имя_реквизита <знак сравнения> Значение, где допускаются знаки сравнения =, <> (не равно), >, >=, <, <= и некоторые другие. Например, Цена > 1000.

2) Условие вида Имя_реквизита_1 <знак сравнения> Имя_реквизита_2. Например, Факт > План.

Имена атрибутов условия выборки должны содержаться в структуре исходного отношения.

Алгебраическая запись выборки имеет вид:

$$T = R[p],$$

где T – результирующее отношение;
 R – исходное отношение;
 p – условие выборки.

Например, выборка $T = F0[D = \langle d3 \rangle]$ приводит к результату

T	A	B	C	D	E
	$a4$	$b1$	$c4$	$d3$	$e3$

Средствами языка SQL для файла $F0$ и условия выборки вида $D = \langle d3 \rangle$ запрос запишется как

SELECT A, B, C, D, E FROM $F0$ WHERE $D = \langle d3 \rangle$ INTO $F1$,

где $F0$ – исходный файл;
 $F1$ – результирующий файл.

Соединение выполняется над двумя исходными отношениями и создает одно результирующее отношение. Каждая строка первого исходного отношения сопоставляется со всеми строками второго исходного отношения. При сопоставлении проверяется условие соединения. Если условие соблюдается, то соответствующие строки двух отношений сцепляются и передаются в результирующее отношение. Условие соединения имеет вид:

Имя_реквизита_1 <знак сравнения> Имя_реквизита_2,
где Имя_реквизита_1 находится в одном исходном отношении, а Имя_реквизита_2 – в другом. Будем использовать следующее обозначение операции соединения:

$$T = R1[p]R2,$$

где T – результирующее отношение;
 $R1$ и $R2$ – исходные отношения;
 p – условие соединения.

Рассмотрим для приведенных ниже отношений $R1$ и $R2$ примеры соединений.

R1	A	B	C
	a1	b1	c1
	a2	b1	c1
	a3	b2	c2
	a1	b2	c1

R2	A	D
	a1	d1
	a2	d2
	a4	d1

1) $W = R1[A = A]R2$

W	A	B	C	A	D
	a1	b1	c1	a1	d1
	a2	b1	c1	a2	d2
	a1	b2	c1	a1	d1

2) $Y = R1[A <> A]R2$

Y	A	B	C	A	D
	a1	b1	c1	a2	d2
	a1	b1	c1	a4	d1
	a2	b1	c1	a1	d1
	a2	b1	c1	a4	d1
	a3	b2	c2	a1	d1
	a3	b2	c2	a2	d2
	a3	b2	c2	a4	d1
	a1	b2	c1	a2	d2
	a1	b2	c1	a4	d1

Соединение называется естественным (натуральным), если:

- знак сравнения – равенство;
- сравнение происходит по всем парам одноименных реквизитов;
- одинаковые имена реквизитов в результате соединения подавляются.

Естественное соединение обозначается $F3 = F1 * F2$ или предложением INNER JOIN в языке SQL.

Рассмотрим отношение *ABCDE* следующего вида.

ABCDE	A	B	C	D	E	F	G	H
	a1	b1	c1	d1	e2	f1	27	32
	a2	b2	c2	d1	e2	f1	44	18
	a3	b3	c3	d2	e5	f1	17	11
	a4	b1	c4	d3	e3	f1	56	61
	a5	b3	c3	d3	e3	f1	43	14
	a6	b4	c2	d7	e6	f6	13	77

Рассмотрим запрос, который выполняет следующие действия. Взять проекцию *ABCDE* на столбцы *B*, *F*, сгруппировать по *B*, *F* и просуммировать значения *H* по сгруппированным строкам. Назовем этот запрос Q1. В форме SQL он будет выглядеть так:
SELECT *ABCDE.B*, *ABCDE.F*, **Sum**(*ABCDE.H*) **AS** *Sum_H*
FROM *ABCDE* **INTO** *BFSumH*
GROUP BY *ABCDE.B*, *ABCDE.F*;
 Синтаксический анализ запроса Q1:
FROM *ABCDE* **INTO** *BFSumH* – означает, что входная таблица есть *ABCDE*, выходная получает название *BFSumH*, **FROM** и **INTO** – служебные слова языка SQL;
SELECT – служебное слово языка SQL, за которым следует перечень столбцов выходной таблицы *BFSumH*;
ABCDE.B, *ABCDE.F* – два столбца выходной таблицы *BFSumH* формируются из указанных столбцов входной таблицы *ABCDE*;
Sum(*ABCDE.H*) **AS** *Sum_H* – третий столбец выходной таблицы *BFSumH* формируется путем суммирования – **Sum**(*ABCDE.H*) – числовых значений столбца *H* входной таблицы *ABCDE* и получает (служебное слово языка SQL **AS**) в выходной таблице *BFSumH* имя *Sum_H*;
GROUP BY *ABCDE.B*, *ABCDE.F* – указывает группировку значений результата по *B* и *F*;
 ; – конец SQL-запроса.

Для построения запроса Q1 средствами экранного сопряжения СУБД Access следует в режиме конструктора СУБД Access выполнить последовательность действий Запрос-создать-конструктор и заполнить экранную форму запроса следующими данными.

Поле:	<i>B</i>	<i>F</i>	<i>H</i>
Имя таблицы:	<i>ABCDE</i>	<i>ABCDE</i>	<i>ABCDE</i>
Групповая операция:	Группировка	Группировка	Sum
Сортировка:			

Задания

Задание 2.1. По данным табл. 2.2 выполните следующие действия:

- убедитесь, что табл. 2.2 удовлетворяет требованиям к нормализованному файлу;

- укажите схему и экземпляр файла;
- для каждого из реквизитов сформулируйте описание или точное определение его домена;
- какие факты, объекты или события отражают строки таблицы?
- в пустой строке 1 дайте буквенное обозначение каждому из реквизитов с учетом того, основание это или признак;
- реализуйте табл. 2.2 средствами СУБД, укажите на машинные средства задания характеристик табл. 2.2.

Таблица 2.2

Файл Бакалея

Номер строки	Код_товара	Вид_товара	Наименование_товара	Расфасовка
1				
2	10100	Чай	Баловень	50 пакетов
3	10150	Чай	Баловень	100 пакетов
4	10200	Чай	Принцесса Нури	Гранулы
5	10300	Чай	Принцесса Гита	50 г
6	10400	Чай	Дилма	100 г
7	10500	Чай	Ахмад	100 г
8	10600	Чай	3 слона	50 г
9	10700	Чай	Пиквик	50 пакетов
10	10800	Чай	Брук Бонд	100 г
11	20090	Кофе	Nescafe	50 г
12	20100	Кофе	Nescafe	100 г
13	20200	Кофе	Nescafe	250 г
14	20300	Кофе	Jacobs	50 г
15	20400	Кофе	Jacobs	100 г
16	20700	Кофе	Pele	100 г
17	20800	Кофе	Monterey	100 г
18	50064	Джем	Белорусский	250 г
19	50300	Конфитюр	Нижегородский	300 г
20	50400	Повидло	Арзамасское	500 г
21	50500	Шоколад	Российский	100 г
22	50900	Сливки	Украинские	5 г

Задание 2.2. Получите (средствами СУБД) проекцию файла F0 (см. табл. 2.1) на столбцы D и E. Реализуйте рассмотренную проекцию SQL-запросом средствами СУБД Access.

Задание 2.3. Пусть имеется файл $F1$ со схемой $CX(F1) = \langle A, B \rangle$ (бинарное отношение) и экземпляром

$$EX(F1) = \{ \langle a1, b1 \rangle; \langle a2, b2 \rangle; \langle a3, b3 \rangle; \langle a4, b1 \rangle; \langle a5, b3 \rangle \}.$$

Постройте проекции $F2 = F1[A]$ и $F3 = F1[B]$.

Получите данный результат с помощью SQL-запроса.

Задание 2.4. Постройте в СУБД Access проекции файла $F0$ (см. табл. 2.1) в следующих вариантах:

- а) все проекции на один реквизит;
- б) все проекции на три реквизита.

Задание 2.5. Постройте в СУБД Access бинарные проекции файла $F0$ (см. табл. 2.1), имеющие:

- а) двухреквизитный ключ;
- б) четыре строки;
- в) три строки.

Докажите, что этот файл не имеет более коротких проекций.

Задание 2.6. Пусть имеется запрос $REL F0 = F0 [E = \langle e2 \rangle]$. Файл $F0$ показан в табл. 2.1. Назовем этот запрос ВКАЕЗ. Отобразите его средствами языка SQL в таблицу. Сделайте это в нескольких вариантах, меняя набор столбцов в операторе SELECT. В необходимых случаях добавьте операцию GROUP BY.

Реализуйте запрос ВКАЕЗ экранными средствами СУБД Access.

Задание 2.7. Реализуйте операции поиска над файлами Бакалея (см. табл. 2.2) и $F0$ (см. табл. 2.1) таким образом, чтобы результирующие файлы содержали:

- 1) 1 запись;
- 2) 2 записи.

Реализуйте различные операции поиска над файлами Бакалея и $F0$, варьируя поисковые условия, проекции и способы задания параметров поиска.

Задание 2.8. Пусть имеется файл $F1$ со схемой $CX(F1) = \langle A, B, C \rangle$ и экземпляром $EX(F1) = \{ \langle a1, b1, c1 \rangle, \langle a2, b1, c2 \rangle \}$.

Выполните проекцию на подсхемы

$$CX(F11) = \langle A, B \rangle;$$

$$CX(F12) = \langle B, C \rangle.$$

Выполните естественное соединение $F2 = F11 * F12$ и убедитесь, что $F1 \neq F2$. Почему это произошло?

Задание 2.9. Пусть имеются отношения $T1$ и $T2$, показанные ниже

$T1$	A	B	C
	$a1$	8	4
	$a2$	1	3
	$a3$	6	5
	$a1$	2	4

$T2$	A	D
	$a1$	5
	$a2$	4
	$a4$	2

Выполните следующие соединения:

- 1) $T3 = T1[B < D]T2$;
- 2) $T4 = T1[C > D]T2$.

Задание 2.10. Получите естественные соединения файлов $F1$ (см. задание 2.8) и $F0$ (см. табл. 2.1), а также отдельных ранее полученных проекций этих файлов.

Проанализируйте случаи естественного соединения без потерь, ловушки связей, декартова произведения, построив соответствующие запросы и таблицы средствами СУБД Access.

Задание 2.11. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.A, ABCDE.B, Sum(ABCDE.G) AS Sum_G
INTO ProjectionABGSum
FROM ABCDE
GROUP BY ABCDE.A, ABCDE.B;
```

Таблица $ABCDE$ приведена в методических указаниях.

Задание 2.12. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ProjectionABGSum.A, BFSumH.B, BFSumH.F INTO
JoinB
FROM BFSumH INNER JOIN ProjectionABGSum ON
BFSumH.B = ProjectionABGSum.B;
```

Таблица $BFSumH$ приведена в методических указаниях.

Задание 2.13. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.A, ABCDE.B, ABCDE.F, ABCDE.E  
FROM ABCDE  
WHERE (((ABCDE.E)=«e2» Or (ABCDE.E)=«e3»));
```

Задание 2.14. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.A, ABCDE.B, Sum(ABCDE.H) AS Sum_H  
FROM ABCDE INTO ABSumH  
GROUP BY ABCDE.B;
```

Задание 2.15. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.B, ABCDE.F, Sum(ABCDE.H) AS Sum_H  
FROM ABCDE  
GROUP BY ABCDE.B;
```

Задание 2.16. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ProjectionABGSum.A, ABSumH.B, ABSumH.A INTO  
JoinA  
FROM ABSumH INNER JOIN ProjectionABGSum ON  
ABSumH.A = ProjectionABGSum.A;
```

Задание 2.17. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.B, ABCDE.F, Avg(ABCDE.H) AS Avg_H  
FROM ABCDE  
GROUP BY ABCDE.B, ABCDE.F;
```

Задание 2.18. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.A, ABCDE.B, ABCDE.F, ABCDE.E  
FROM ABCDE  
WHERE (((ABCDE.E)=«e3» Or (ABCDE.F)=«f1»));
```

Задание 2.19. Реализуйте следующий SQL-запрос. Постройте также этот запрос средствами экранного сопряжения СУБД Access.

```
SELECT ABCDE.A, ABCDE.B, ABCDE.F, ABCDE.E  
FROM ABCDE  
WHERE (((ABCDE.E)=«e3» And (ABCDE.F)=«f1»));
```

Задание 2.20. Сведения об учебном процессе зафиксированы в четырех отношениях:

Студ(Гр, Зач, ФИО);

Оценка(Гр, Зач, Дисц, Дата, Пр, Оц);

Расп(Дата, Гр, Дисц, Пр);

Преп(Дисц, Пр, Каф).

В задании используются следующие обозначения:

Студ – студент;

Гр – номер группы;

Зач – номер зачетной книжки;

ФИО – фамилия студента;

Дисц – дисциплина;

Пр – фамилия преподавателя;

Оц – оценка;

Расп – расписание;

Преп – преподаватель;

Каф – название кафедры.

Запишите с помощью операторов реляционной алгебры следующие запросы. В тех случаях, когда это возможно, запишите запросы на языках Visual FoxPro и SQL.

1. Найти фамилии преподавателей, ведущих занятия в группах 305 и 306 одновременно.

2. Какие оценки получил студент Федоров?

3. У каких студентов преподает Иванов?

4. Какие студенты сдали те же экзамены, что и Федоров?

5. Какие преподаватели работают 10.04.04?

6. Какие преподаватели ведут занятия в тех же группах, что и Иванов?

7. По каким предметам сдается зачет, а не экзамен?

8. Какие студенты изучают дисциплину «высшая математика» 10.04.04?

9. Какие дисциплины преподаются на Кафедре высшей математики?

10. Какие преподаватели преподают дисциплину «высшая математика»?

11. Какие преподаватели поставили удовлетворительные оценки в группе 305?

12. Какие экзамены сданы у всех студентов группы 305?

13. Какие кафедры ведут занятия в группе 305?

14. Какие преподаватели работают в те же дни, что и Иванов?

15. Какие преподаватели поставили отличные оценки студенту Федорову?

16. По каким дисциплинам студент Федоров получил отличные оценки?

17. Какие студенты учатся в той же группе, что и Федоров?

2.2.

Функциональные зависимости и ключи

Методические указания

Функциональные зависимости определяются для реквизитов, находящихся в одном и том же нормализованном отношении.

В простейшем случае в функциональной зависимости участвуют два реквизита. В отношении $R(A, B, \dots)$ реквизит A функционально определяет реквизит B , если в любой момент времени каждому значению A соответствует единственное значение B (обозначается $A \rightarrow B$).

Иначе говорят, что B функционально зависит от A (обозначается $B = f(A)$). Первое обозначение оказывается более удобным, когда число функциональных зависимостей растет и их взаимосвязи становятся труднообозримыми; оно и будет использоваться в дальнейшем. Отсутствие функциональной зависимости обозначается $A \not\rightarrow B$.

Рассмотрим простой пример с реквизитами ФИО и ГР (год рождения) в отношении $R1$.

Предположим, что в столбце ФИО представлены сведения о разных людях и соответствующие значения в столбце не повторяются. Тогда можно говорить о наличии функциональной зависимости $\text{ФИО} \rightarrow \text{ГР}$, поскольку каждому значению реквизита ФИО в отношении $R1$ соответствует единственное значение реквизита ГР. Можно утверждать, что это ограничение будет соблю-

даться и далее, так как оно перефразируется в утверждение: «у каждого человека единственный год рождения», которое справедливо.

R1	ФИО	ГР
	Иванов	1980
	Петров	1979
	Сидоров	1980
	Яшина	1978

Практически каждое ограничение для проверки функциональной зависимости можно преобразовать в утверждение о свойствах объектов предметной области, которое можно проверить, не анализируя множество значений соответствующего отношения. Именно так мы и будем поступать в дальнейшем. Наличие в столбце ГР повторяющихся годов не опровергает установленной нами зависимости, но это означает $\text{ГР} \not\rightarrow \text{ФИО}$.

Одновременное соблюдение двух зависимостей вида $A \rightarrow B$ и $B \rightarrow A$ называется взаимно-однозначным соответствием и обозначается $A \leftrightarrow B$.

В качестве примера рассмотрим отношение R2 с реквизитами Магазин и Расч (номер расчетного счета).

R2	Магазин	Расч
	«Океан»	770019
	«Янтарь»	770079

Можно утверждать, что у каждого магазина единственный номер расчетного счета и каждый расчетный счет принадлежит единственному магазину. Это доказывает справедливость функциональных зависимостей $\text{Магазин} \rightarrow \text{Расч}$ и $\text{Расч} \rightarrow \text{Магазин}$, т.е. $\text{Магазин} \leftrightarrow \text{Расч}$.

Наконец, самыми распространенными являются случаи отсутствия функциональных зависимостей, например, $\text{ФИО} \not\rightarrow \text{Дисциплина}$ и $\text{Дисциплина} \not\rightarrow \text{ФИО}$ в отношении R3, описывающем экзамены студентов. Здесь каждый студент сдает экзамены по нескольким дисциплинам, и по каждой дисциплине экзамен сдается многими студентами.

R3	ФИО	Дисциплина
	Иванов	Информатика
	Яшина	Информатика
	Сидоров	Физика
	Яшина	Физика

Таким образом, для реквизитов A и B некоторого отношения возможны следующие ситуации:

- отсутствие функциональной зависимости;
- наличие $A \rightarrow B$ (или $B \rightarrow A$), но не обе зависимости вместе;
- наличие взаимно-однозначного соответствия $A \leftrightarrow B$.

Понятие «функциональная зависимость» распространяется на ситуацию с тремя и более реквизитами в следующей форме. Группа реквизитов (для определенности A, B, C) функционально определяет реквизит D в отношении $T(A, B, C, D, \dots)$, если каждому сочетанию значений $\langle a, b, c \rangle$ соответствует единственное значение d (a – значение A , b – значение B , c – значение C , d – значение D). Наличие такой функциональной зависимости будем обозначать $A, B, C \rightarrow D$. Случай, когда в правой части функциональной зависимости имеется несколько реквизитов, не нуждается в специальном рассмотрении.

Пусть в отношении $T1$ представлены сведения о закончившихся экзаменах $T1(\text{ФИО}, \text{Дата}, \text{Дисциплина}, \text{Преподаватель}, \text{Оценка})$. ФИО означает фамилию, имя, отчество студента.

Ограничение, состоящее в том, что студент не может в один день сдать два и более экзаменов, означает справедливость ряда функциональных зависимостей:

ФИО, Дата \rightarrow Дисциплина;

ФИО, Дата \rightarrow Преподаватель;

ФИО, Дата \rightarrow Оценка.

Наличие функциональных зависимостей связано с применяемыми способами кодирования реквизитов. Так, для множества учреждений можно утверждать, что каждый отдел (как объект предметной области) относится к единственному учреждению. Однако этого недостаточно для доказательства функциональной зависимости Отдел \rightarrow Учреждение. Если в каждом учреждении отделы нумеруются последовательно, начиная с 1, то функциональная зависимость неверна. Если же код отдела кроме номера содержит и код учреждения (или уникальность кодов обеспечивается каким-то другим способом), то функциональная зависимость Отдел \rightarrow Учреждение справедлива.

Зависимость ФИО \rightarrow ГР в $R1$ соблюдается, если ФИО является реквизитом-идентификатором для каждого человека, что может быть справедливо только для небольших множеств людей. Невнимание к способам кодирования реквизитов может привести к несоответствию функциональных зависимостей и хранящихся данных, что является серьезной проектной ошибкой.

Для показателя с множеством реквизитов-признаков $P = \{P_1, P_2, \dots, P_n\}$ и реквизитом-основанием Q справедлива функциональная зависимость $P \rightarrow Q$, хотя нельзя утверждать, что это единственная зависимость на указанных реквизитах.

С помощью функциональных зависимостей определяется понятие «ключ» отношения, точнее, ряд разновидностей ключей – вероятные, первичные и вторичные.

Вероятным ключом отношения называется такое множество реквизитов, что каждое сочетание их значений встречается только в одной строке отношения, и никакое подмножество реквизитов этим свойством не обладает. Вероятных ключей в отношении может быть несколько.

Рассмотрим в качестве примера отношение $T4$ (имена и значения реквизитов – условные).

$T4$	ZEN	RAM	AST	SPIM	BIG
	1A	31	DWA	NII	73
	3B	01	BUN	CUP	16
	3D	30	MUN	LAW	58
	4D	31	SAB	NII	40
	7B	30	SAB	IRT	38

Рассмотрим ряд функциональных зависимостей.

1. $ZEN \rightarrow BIG$.

Эта функциональная зависимость верна поскольку атрибут ZEN – вероятный ключ отношения.

2. $RAM \rightarrow AST$.

Эта функциональная зависимость не верна в соответствии с сочетаниями значений $\langle 31, DWA \rangle$ и $\langle 31, SAB \rangle$.

3. $RAM, AST \rightarrow BIG$.

Эта функциональная зависимость верна поскольку атрибуты RAM, AST вместе образуют вероятный ключ отношения.

4. $RAM, SPIM \rightarrow BIG$.

Эта функциональная зависимость не верна в соответствии с сочетаниями значений $\langle 31, NII, 73 \rangle$ и $\langle 31, NII, 40 \rangle$.

Можно утверждать, что вероятным ключом отношения T_4 является реквизит ZEN (значения в столбце ZEN не повторяются). Кроме того, еще один вероятный ключ представлен парой реквизитов RAM, AST.

Роль вероятных ключей при обработке данных определяется тем, что выборка по известному значению вероятного ключа дает в результате одну строку отношения либо ни одной.

На практике реквизиты вероятного ключа отношения связываются со свойствами тех объектов и событий, информация о которых хранится в отношении. Если в результате корректировки отношения изменились имена реквизитов, образующих ключ, то это свидетельствует о серьезном искажении информации. Следовательно, систематическая проверка свойств вероятного ключа позволяет следить за достоверностью информации в отношении. Если в отношении присутствует несколько вероятных ключей, одновременно следить за ними очень трудно и целесообразно выбрать один из них в качестве основного (первичного).

Первичным ключом отношения называется такой вероятный ключ, по значениям которого проводится контроль достоверности информации в отношении. Применительно к экономической информации в подавляющем большинстве случаев отношения, полученные из экономических документов, содержат единственный вероятный ключ, который является и первичным ключом. Это объясняется тем, что содержимое экономических документов понимается всеми пользователями одинаково. Далее будем иметь в виду только такие отношения. Наличие двух и более вероятных ключей в отношениях с осмысленной информацией можно объяснить наличием нескольких возможных способов интерпретации одних и тех же данных. Первичный ключ часто называется просто «ключ».

В отношениях с большим числом строк поиск первичного ключа путем непосредственного применения определения достаточно затруднен. Кроме того, на стадии проектирования экономической информационной системы значения многих отношений обычно неизвестны. Поэтому практически первичный ключ отношения определяется по известным функциональным зависимостям.

Каждое значение первичного ключа встречается только в одной строке отношения. Значение любого реквизита в этой строке также единственное. Если обозначить K реквизиты первичного ключа в отношении $R(A, B, C, \dots, J)$, то справедливы функциональ-

ные зависимости: $K \rightarrow A, K \rightarrow B, \dots, K \rightarrow J$. Иными словами, набор реквизитов первичного ключа функционально определяет любой реквизит отношения. Обратное утверждение также верно – если найдена группа реквизитов, которая функционально определяет все реквизиты отношения по отдельности и эту группу нельзя сократить, то найден первичный ключ отношения.

Вернемся к отношению $T1$ с функциональными зависимостями:

ФИО, Дата \rightarrow Дисциплина;

ФИО, Дата \rightarrow Преподаватель;

ФИО, Дата \rightarrow Оценка.

Нетрудно установить, что

ФИО, Дата \rightarrow ФИО (в каждом сочетании значений ФИО, Дата значение ФИО встречается один раз);

ФИО, Дата \rightarrow Дата.

Следовательно, первичный ключ в отношении $T1$ составляют реквизиты ФИО, Дата и при поиске ключа не потребовались конкретные значения $T1$.

Знание ключа отношения позволяет устанавливать ряд функциональных зависимостей, например, в $T4$ – это $ZEN \rightarrow BIG, RAM, AST \rightarrow BIG$.

Для множества функциональных зависимостей известен ряд закономерностей, которые выражаются теоремами. Знание теорем позволяет из исходного множества функциональных зависимостей получать производные зависимости.

Приведем ряд теорем о функциональных зависимостях. Реквизиты, фигурирующие в каждой теореме, должны находиться в одном и том же отношении.

Теорема 1

$$A, B \rightarrow A \text{ и } A, B \rightarrow B.$$

Теорема 2

$$A \rightarrow B \text{ и } A \rightarrow C \text{ тогда и только тогда, когда } A \rightarrow BC.$$

Теорема 3

$$\text{Если } A \rightarrow B \text{ и } B \rightarrow C, \text{ то } A \rightarrow C.$$

Теорема 4

$$\text{Если } A \rightarrow B, \text{ то } AC \rightarrow B \text{ (} C \text{ произвольно).}$$

Теорема 5

Если $A \rightarrow B$, то $AC \rightarrow BC$ (C произвольно).

Теорема 6

Если $A \rightarrow B$ и $BC \rightarrow D$, то $AC \rightarrow D$.

Для некоторого множества функциональных зависимостей F введем множество $F\sim$, называемое покрытием. Покрытие $F\sim$ содержит все функциональные зависимости, которые можно получить из множества F в результате применения теорем Т1–Т6 (включая и содержащее F). Одно и то же покрытие $F\sim$ может быть получено из различных множеств функциональных зависимостей. Среди таких множеств выделим множество с минимальным числом зависимостей и назовем его минимальным покрытием (базисом) множества зависимостей F . Можно сказать, что минимальным покрытием называется множество функциональных зависимостей, из которого удалены все зависимости, являющиеся следствиями оставшихся зависимостей и теорем Т1 – Т6.

Непосредственное применение теорем к множеству F целесообразно только при небольшом количестве зависимостей. Теоретической основой методов поиска минимального покрытия служит аппарат булевых функций.

Можно рекомендовать следующие теоремы:

- если $A \rightarrow BC$, то $A \rightarrow B$ и $A \rightarrow C$;
- если $A \rightarrow B$, то $AD \rightarrow B$;
- если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$.

Зависимости, указанные в условии теоремы, остаются в списке функциональных зависимостей, а зависимости, указанные в заключении теоремы, удаляются.

Если заранее известно, что вероятный ключ в отношении только один, то его можно найти простым способом. Вероятный ключ (если он единственный, то совпадает с первичным ключом) – это набор реквизитов, которые не встречаются в правых частях всех функциональных зависимостей. Фактически, из полного списка реквизитов отношения необходимо вычеркнуть реквизиты, встречающиеся в правых частях всех функциональных зависимостей. Оставшиеся реквизиты образуют первичный ключ.

Последний вопрос, нуждающийся в рассмотрении, – это наличие в отношениях реляционной базы данных неопределенных (нулевых) значений. Безопасной является ситуация, когда неопределенные значения отсутствуют у реквизитов, входящих в левые

части каких-либо функциональных зависимостей. Если это требование не соблюдается, то операция выборки может создавать результирующие отношения с неопределенными значениями в тех позициях, где их, по сути, быть не должно, а операция соединения может давать полностью неправильные результаты.

Задания

Задание 2.21. Найдите все функциональные зависимости и вероятные ключи в файлах Бакалея (см. табл. 2.2), F0 (см. табл. 2.1), T4 (см. методические указания).

Задание 2.22. Ниже приведен справочник кодов субъектов Российской Федерации (табл. 2.3). Эти коды применяются при регистрации автотранспортных средств в органах ГИБДД (ГАИ), при формировании идентификационных номеров налогоплательщиков и в ряде других систем кодирования. Выполните следующие действия:

- создайте фрагмент таблицы **Коды субъектов Российской Федерации**, содержащий не менее 16 строк, укажите наличие функциональной зависимости, реализуйте таблицу средствами СУБД Access;

- выясните, приводит ли к нарушению обнаруженной функциональной зависимости пополнение таблицы **Коды субъектов Российской Федерации** новыми кодами: 90 – для Московской области, 99 и 97 – для Москвы, 95 – для Чеченской Республики. Убедитесь в правильности вашего ответа путем ввода соответствующих изменений в машинную версию таблицы и последующей проверкой.

Т а б л и ц а 2.3

Коды субъектов Российской Федерации

Код	Название субъекта
1	Республика Адыгея
2	Республика Башкортостан
3	Республика Бурятия
4	Республика Алтай
5	Республика Дагестан
6	Республика Ингушетия

Код	Название субъекта
7	Кабардино-Балкарская Республика
8	Республика Калмыкия
9	Карачаево-Черкесская Республика
10	Республика Карелия
11	Республика Коми
12	Республика Марий Эл
13	Республика Мордовия
14	Республика Саха (Якутия)
15	Республика Северная Осетия – Алания
16	Республика Татарстан
17	Республика Тыва
18	Удмуртская Республика
19	Республика Хакасия
20	Чеченская Республика
21	Чувашская Республика
22	Краснодарский край
23	Красноярский край
24	Приморский край
25	Ставропольский край
26	Хабаровский край
27	Амурская область
28	Архангельская область
29	Астраханская область
30	Белгородская область
31	Брянская область
32	Владимирская область
33	Волгоградская область
34	Вологодская область
35	Воронежская область
36	Ивановская область
37	Иркутская область
38	Калининградская область
39	Калужская область
40	Камчатская область
41	Кемеровская область

Код	Название субъекта
42	Кировская область
43	Костромская область
44	Курганская область
45	Курская область
46	Ленинградская область
47	Липецкая область
48	Магаданская область
49	Московская область
50	Мурманская область
51	Нижегородская область
52	Новгородская область
53	Новосибирская область
54	Омская область
55	Оренбургская область
56	Орловская область
57	Пензенская область
58	Пермская область
59	Псковская область
60	Ростовская область
61	Рязанская область
62	Самарская область
63	Саратовская область
64	Сахалинская область
65	Свердловская область
66	Смоленская область
67	Тамбовская область
68	Тверская область
69	Томская область
70	Тульская область
71	Тюменская область
72	Ульяновская область
73	Челябинская область
74	Читинская область
75	Ярославская область
76	Москва
77	Санкт-Петербург

Код	Название субъекта
78	Еврейская автономная область
79	Коми-Пермяцкий автономный округ
80	Корякский автономный округ
81	Ненецкий автономный округ
82	Таймырский (Долгано-Ненецкий) автономный округ
83	Усть-Ордынский Бурятский автономный округ
84	Ханты-Мансийский автономный округ
85	Чукотский автономный округ
86	Эвенкийский автономный округ
87	Ямало-Ненецкий автономный округ

Задание 2.23. Средствами СУБД Access создайте дополнительно к таблице **Коды субъектов Российской Федерации** (табл. 2.3) структуру, позволяющую классифицировать субъекты Федерации по типу (край, республика, область и т.д.). Какие новые функциональные зависимости появятся? Найдите все вероятные ключи. Создайте следующие запросы:

- найти субъект Федерации по коду;
- определить тип заданного субъекта Федерации;
- выделить все субъекты Федерации заданного типа.

Реализуйте запросы средствами SQL.

Задание 2.24. Рассмотрите отношение *W2*.

<i>W2</i>	Продукция	Цена продукции	Комплектуемое изделие
	Эдв-12	40	Вк-15
	Эдв-12	40	Р-20
	Эдв-30	20	Вк-15
	Эдв-30	20	Р-20
	ЗВИ	120	Р-20

1. Постройте не менее двух вариантов SQL-запросов, результатом которых служили бы двухреквизитные проекции отношения *W2*.

2. Укажите функциональные зависимости отношения *W2*.

3. Укажите вероятный ключ отношения *W2*.

Задание 2.25. Предложите алгоритм проверки наличия функциональной зависимости для двухреквизитного отношения. Реализуйте этот алгоритм средствами известного вам языка программирования.

Задание 2.26. Докажите или опровергните наличие функциональных зависимостей в следующем отношении.

T5	KROL	FOUN	DIN	DOT	IMH
	54	32	LO	KU	V3
	34	78	OL	VD	Z2
	45	27	KE	KE	R9
	94	52	LO	VP	P3
	54	65	KE	WA	E7
	45	32	OL	AD	V3
	34	78	KE	WA	Z2
	94	85	OL	KI	A8

1. $DIN \rightarrow KROL$.
2. $IMH \rightarrow KROL$.
3. $FOUN \rightarrow IMH$.
4. $KROL \rightarrow FOUN$.
5. $DOT, KROL \rightarrow DIN$.

Задание 2.27. Докажите или опровергните наличие функциональных зависимостей в следующем отношении.

T6	POR	TRIG	CHEK	KOV	SKB
	RA	2T	F4	GN	BX
	XO	1A	E7	NI	WX
	CW	9X	G8	VD	AX
	KP	5E	U6	QW	BX
	WA	9X	E7	EP	RX
	RA	5E	U6	KA	XY
	AV	1A	G8	KR	AX
	WE	9X	F4	EP	BX

1. $POR \rightarrow KOV$.
2. $CHEK \rightarrow POR$.
3. $TRIG \rightarrow CHEK$.
4. $POR, CHEK \rightarrow KOV$.
5. $SKB, KOV \rightarrow TRIG$.

Задание 2.28. Докажите или опровергните наличие функциональных зависимостей в следующем отношении.

T7	INES	KATR	LID	ELIT	SMID
	LU	01	35	UW	70
	GN	02	54	WA	60
	WA	03	35	YW	70
	KU	01	35	GO	40
	LU	02	60	AR	60
	GN	03	35	CE	20
	WA	01	35	SS	30
	AR	02	54	DP	10

1. ELIT \rightarrow SMID.
2. LID \rightarrow KATR.
3. SMID \rightarrow INES.
4. INES, ELIT \rightarrow LID.
5. KATR, LID \rightarrow SMID.

Задание 2.29. Докажите или опровергните наличие функциональных зависимостей в следующем отношении.

T8	KOS	KOGR	WKII	DUBL	PRIN
	AK	61	WA	570	WS
	AT	44	PR	751	WA
	AK	20	AI	856	RI
	AU	73	WA	536	TX
	AT	15	PR	751	AX
	AK	94	FY	456	PI
	AU	30	CM	846	CI
	AT	52	VD	751	WS

1. KOS \rightarrow KOGR.
2. KOGR \rightarrow WKII.
3. DUBL, PRIN \rightarrow WKII.
4. WKII \rightarrow PRIN.
5. KOS, WKII \rightarrow DUBL.

Задание 2.30. Укажите функциональные зависимости в следующих отношениях. Найдите вероятные ключи.

T0(ФИО, отдел, руководитель отдела);
T1(ФИО, отдел, должность);
T2(отдел, проект, число исполнителей);
T3(электродвигатель, мощность, покупатель, купленное количество);
T4(преподаватель, кафедра, факультет);
T5(автор, статья, журнал);
T6(преподаватель, кафедра, дисциплина);
T7(автомобиль, грузоподъемность, расход топлива);
T8(отрасль, предприятие, бригада);
T9(изделие, деталь, вес изделия).

Задание 2.31. Укажите следствия из множества функциональных зависимостей $\{A \rightarrow B, B \rightarrow C\}$.

Задание 2.32. Укажите следствия из множества функциональных зависимостей $\{X \rightarrow X1, X \rightarrow X2, X \rightarrow X3, X \rightarrow X4\}$.

2.3.

Нормальные формы отношений

Методические указания

Нормальные формы отношений – это отношения с дополнительно соблюдаемыми ограничениями. Нормальные формы нумеруются, и чем больше номер нормальной формы, тем больше ограничений должно соблюдаться.

Нормализованный файл соответствует определению отношения в первой нормальной форме (1НФ).

Отношение находится во второй нормальной форме (2НФ), если оно соответствует 1НФ и не содержит неполных функциональных зависимостей.

Неполная функциональная зависимость – это две зависимости:

- вероятный ключ отношения функционально определяет некоторый неключевой реквизит;
- часть вероятного ключа функционально определяет этот же неключевой реквизит.

Отношение, не соответствующее 2НФ, характеризуется избыточностью хранимых данных. В качестве примера рассмотрим

функциональные зависимости отношения $TT(\text{Магазин, Изделие, Цена, План_текущего_года})$:

- 1) Магазин, Изделие \rightarrow План_текущего_года;
- 2) Изделие \rightarrow Цена.

Вероятным ключом в TT являются реквизиты Магазин, Изделие. Для доказательства можно сослаться на функциональные зависимости:

Магазин, Изделие \rightarrow План_текущего_года;

- 3) Магазин, Изделие \rightarrow Цена (теорема Т4);
- 4) Магазин, Изделие \rightarrow Магазин (теорема Т1);
- 5) Магазин, Изделие \rightarrow Изделие (теорема Т1).

Зависимости 3) и 2) образуют неполную функциональную зависимость, по этой причине отношение TT находится лишь в 1НФ, а не в 2НФ.

Избыточность иллюстрируется тем фактом, что цена изделия указывается столько раз, сколько магазинов продают это изделие. Переход к 2НФ и соответственно устранение отмеченной избыточности данных связано с созданием двух отношений вместо исходного отношения TT

$TT1 = TT[\text{Магазин, Изделие, План_текущего_года}]$,

$TT2 = TT[\text{Изделие, Цена}]$.

Ключом в $TT1$ служат реквизиты Магазин, Изделие, в $TT2$ – Изделие, и легко определить, что оба отношения соответствуют требованиям 2НФ.

База данных находится в 2НФ, если все ее отношения находятся в 2НФ.

Отношение соответствует 3НФ, если оно соответствует 2НФ и среди его реквизитов отсутствуют транзитивные функциональные зависимости (ФЗ). Транзитивная ФЗ – это две ФЗ:

- вероятный ключ отношения функционально определяет неключевой реквизит;
- этот реквизит функционально определяет другой неключевой реквизит.

Если K – ключ отношения, A, B – неключевые реквизиты и справедливы ФЗ $K \rightarrow A, A \rightarrow B$, то они являются транзитивными. Частный случай транзитивной ФЗ – неполная ФЗ, когда $K = C, E$ и $K \rightarrow E, E \rightarrow A$.

Рассмотрим, например, функциональные зависимости отношения $P(\text{ФИО, Группа, Факультет})$

- 6) ФИО \rightarrow Группа.

7) Группа \rightarrow Факультет.

8) ФИО \rightarrow Факультет.

Ключ отношения R – ФИО. Зависимости 6) и 7) образуют транзитивную ФЗ, поэтому R находится в 2НФ, а не в 3НФ.

Избыточность данных в R связана с тем, что принадлежность группы к факультету указывается столько раз, сколько студентов обучается в этой группе.

Переход от R к отношениям в 3НФ дает следующие результаты:

$R1 = P[\text{ФИО}, \text{Группа}]$

$R2 = P[\text{Группа}, \text{Факультет}]$.

Отношения $R1$, $R2$ получились двухреквизитными, поэтому нарушение требований 3НФ в них невозможно.

База данных находится в 3НФ, если все ее отношения находятся в 3НФ.

Приведенные примеры показывают, что отношения, в которых соблюдается одна ФЗ либо ни одной, будут соответствовать условиям 2НФ и 3НФ, так как неполная и транзитивная ФЗ представляют собой две зависимости. На этом принципе основан алгоритм получения отношений в 3НФ.

Исходными данными для алгоритма служит некоторый список реквизитов, охватывающий одно отношение, базу данных или ее часть. В любом случае предполагается (хотя бы теоретически) наличие одного отношения с заданным списком реквизитов. В противном случае нельзя применять некоторые теоремы о функциональных зависимостях и нельзя гарантировать, что одна и та же ФЗ, например, $A \rightarrow B$, справедлива в двух различных отношениях R и S , соответствует равным отношениям $R[A, B]$ и $S[A, B]$.

Алгоритм получения отношений в 3НФ обладает следующими свойствами:

- сохраняет все первоначальные функциональные зависимости, т.е. зависимость, справедливая в R , справедлива и в одном из производных отношений. Это гарантирует получение осмысленных отношений с легко интерпретируемой структурой;

- обеспечивает соединение без потерь, т.е. значения исходного отношения R могут быть восстановлены из проекций отношения R с помощью операции соединения;

- результат декомпозиции в 3НФ обычно содержит меньше значений реквизитов, чем исходное отношение R (происходит уменьшение избыточности).

Алгоритм нормализации к 3НФ

Исходные данные – множество всех реквизитов базы данных.

Метод – создание отношений, в которых соблюдается одна функциональная зависимость либо ни одной.

Для реализации метода необходимо выполнить следующие шаги.

Шаг 1. Получить исходное множество функциональных зависимостей для реквизитов рассматриваемой базы данных.

Если исходные функциональные зависимости не удастся определить путем анализа содержательных ролей реквизитов, приходится использовать перечисление и отбраковку допустимых вариантов функциональных зависимостей.

Для этого рассматриваются все сочетания по два реквизита и в каждом случае доказывается или отвергается функциональная зависимость. Затем рассматриваются сочетания:

- по три реквизита, где первые два могут функционально определять третий;
- по четыре реквизита, где первые три могут функционально определять четвертый и т.д.

Применение теорем о функциональных зависимостях позволяет сократить количество рассматриваемых вариантов. Практически перечисление вариантов заканчивается, когда сочетания реквизитов станут содержать первичный ключ.

Шаг 2. Получить минимальное покрытие множества функциональных зависимостей. В минимальном покрытии должны отсутствовать зависимости, которые являются следствием оставшихся зависимостей по теоремам Т1 – Т6. В частности, требуется объединить функциональные зависимости с одинаковой левой частью в одну зависимость. Обозначим полученное минимальное покрытие функциональных зависимостей через $F = \{f_1, \dots, f_i, \dots, f_k\}$.

Шаг 3. Определить первичный ключ отношения.

Шаг 4. Для каждой функциональной зависимости f_i создать проекцию исходного отношения $R_i = R[X_i]$, где X_i – объединение реквизитов из левой и правой частей f_i .

Шаг 5. Если первичный ключ исходного отношения не вошел полностью ни в одну проекцию, полученную на шаге 4, необходимо создать отдельное отношение из реквизитов ключа.

Для практического применения алгоритма нормализации до 3НФ необходимо решить два вопроса:

- как учесть наличие взаимно-однозначных соответствий?
- как сократить объем перебора вариантов при первоначальном определении множества функциональных зависимостей?

Для взаимно-однозначных соответствий принято выделение старшего (по объему представляемого понятия) реквизита, который затем представляет все реквизиты взаимно-однозначного соответствия.

Для ускорения шага 1 алгоритма нормализации не рассматриваются такие зависимости, которые являются следствием уже найденных зависимостей и теорем Т1–Т6, и используется ряд закономерностей об отсутствии функциональных зависимостей. Среди них теорема:

Если $A \rightarrow B$ и $D \not\rightarrow C$, то $ABD \not\rightarrow C$

и правило, согласно которому функциональные зависимости с реквизитом-основанием в левой части не требуется рассматривать (в противном случае алгоритм станет создавать отношения, состоящие только из реквизитов-оснований, не имеющие экономического смысла).

Задания

Задание 2.33. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
ФИО вкладчика (фио)	номер \rightarrow фио
Номер сберкнижки (номер)	номер, дата \rightarrow расход
Дата	номер, дата \rightarrow остаток
Приход	Номер, дата \rightarrow приход, расход, остаток
Расход	Номер, дата \rightarrow приход
Остаток	

Задание 2.34. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

Цех

Год

Код_станка

количество станков (кол-во)

код_детали

план производства деталей

Функциональные зависимости

цех, код_станка → кол-во

цех, код_станка, код_детали → кол-во

цех, год, код_детали → план

цех, код_станка, год → кол-во

Задание 2.35. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

фио служащего (фио)

должность

Дата

Зарплата

Имя_ребенка

Возраст ребенка

Функциональные зависимости

фио, дата → должность

фио → должность

фио, дата → зарплата

фио, имя_ребенка → возраст

Задание 2.36. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

Табельный номер (таб. №)

Фамилия рабочего (фио)

Цех

Участок

Дата

Сумма зарплаты (сумма)

Функциональные зависимости

участок → цех

таб. № → цех

таб. № → участок

таб. №, дата → сумма

таб. № → фио

таб. № → фио, участок

Задание 2.37. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Магазин	книга, магазин → издательство
Книга	магазин, книга, дата → кол-во
Цена	книга → цена
Издательство	книга → цена, издательство
Дата	книга → издательство
Количество продано (кол-во)	

Задание 2.38. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Отправитель	получатель, изделие → адрес
Получатель	отправитель, получатель, изделие → кол-во
Адрес получателя (адрес)	получатель → адрес
Изделие	отправитель, изделие → цена
Цена	изделие → цена
Кол-во на месяц (кол-во)	

Задание 2.39. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Аэропорт отправления	№ рейса, дата → № самолета
номер рейса (№ рейса)	№ самолета → кол-во
Количество мест (кол-во)	№ рейса → пункт назначения
Бортовой № самолета	№ рейса → пункт назначения, аэропорт
Пункт назначения	№ рейса → аэропорт
Дата вылета (дата)	

Задание 2.40. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

фио студента
 дата поступления в вуз
 факультет
 группа
 место работы
 Зарплата

Функциональные зависимости

фио → группа
 фио → факультет
 фио → дата
 группа → факультет
 место работы → зарплата
 фио → группа, дата

Задание 2.41. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

№ больницы
 № палаты
 Пациент
 Домашний адрес
 Лечащий врач (врач)
 Диагноз

Функциональные зависимости

№ больницы, пациент → № палаты
 № больницы, пациент → врач
 пациент → домашний адрес
 № больницы, пациент → № палаты, врач

Задание 2.42. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

дата матча (дата)
 Команда-хозяин
 Команда-гость
 счет матча
 число очков в чемпионате

Функциональные зависимости

дата, команда-хозяин → счет
 дата, команда-хозяин → число очков
 дата, команда-хозяин → счет, число очков
 дата, команда-хозяин, команда-гость → счет

Задание 2.43. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

ФИО студента (фио)
 Дата поступления в вуз (дата)
 Факультет
 Группа
 Общественная работа

Функциональные зависимости

фио → группа
 фио → факультет
 фио → дата
 Группа → факультет
 фио → группа, дата

Задание 2.44. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Фιο служащего (фιο)	фιο → отдел
Тема работы	фιο → учреждение
Источник финансирования темы (фин)	тема работы → фин
Отдел	отдел → учреждение
Учреждение	фιο, отдел → учреждение

Задание 2.45. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Порт	Судно, дата отплытия → порт
Судно	Судно, дата отплытия → порт назначения
грузоподъемность	Судно, дата отплытия → порт, порт назначения
дата отплытия	Судно → грузоподъемность
порт назначения	

Задание 2.46. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты	Функциональные зависимости
Дисциплина	время, аудитория → дисциплина
Преподаватель	время, преподаватель, дисциплина → аудитория
время занятий	время, преподаватель → аудитория
Аудитория	время, фιο → аудитория
фιο студента (фιο)	

Задание 2.47. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

Учреждение

Отдел

Тема

код оборудования

фио сотрудника (фио)

Продолжительность работы

Функциональные зависимости

фио → учреждение

фио → отдел

код оборудования → отдел

отдел → учреждение

тема, фио → продолжительность

код оборудования → учреждение

Задание 2.48. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

Отдел

фио сотрудника (фио)

номер комнаты

Телефон

тема работы

Продолжительность работы

Функциональные зависимости

фио → отдел

телефон → отдел

телефон → номер комнаты

фио, тема → продолжительность

номер комнаты → отдел

Задание 2.49. Примените алгоритм получения отношений в 3НФ, если 3НФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

Реквизиты

Завод

Продукция

Комплектуемое изделие

Цена продукции

Выпуск продукции за год

Функциональные зависимости

продукция → цена

завод, продукция → выпуск

завод, продукция → цена

Задание 2.50. Рассмотрите отношение $ABT(P1, P2, Q3)$. Это отношение описывает городские автобусные маршруты, т.е. содержит количество строк по числу автобусных маршрутов. $P1$ обозначает номер автобусного маршрута, $P2$ – марка автобуса, $Q3$ – вместимость автобуса.

<i>ABT</i>	<i>P1</i>	<i>P2</i>	<i>Q3</i>
	18	ЛАЗ-677	42
	28	ЛАЗ-677	42
	26	Икарус 260	56
	30	ЛИАЗ-684	45
	507	ПАЗ-672	21

Пусть *P1* (номер автобусного маршрута) – ключ, $P1 \rightarrow P2$. Рассмотрите отношение $ABT1 = ABT[P2, Q3]$, содержащее вместимость для каждой марки автобуса. Здесь имеет место $P2 \rightarrow Q3$. Операции проецирования и естественного соединения позволяют выбрать вариант файловой структуры при создании реляционной базы данных. Возможны два варианта.

1. Создаем отношение *ABT* со схемой $CX(ABT) = \langle P1, P2, Q3 \rangle$.
2. Создаем отношения *ABT1*, *ABT2* со схемами $CX(ABT1) = \langle P2, Q3 \rangle$ и $CX(ABT2) = \langle P1, P2 \rangle$ соответственно являющиеся проекциями отношения *ABT*. При этом для получения отношения *ABT* необходимо использовать операцию естественного соединения отношений *ABT1* и *ABT2*.

Выполните:

- создайте отношение *ABT* средствами СУБД;
- создайте запросы с помощью SQL, формирующие указанные проекции *ABT1* и *ABT2*;
- выполните естественное соединение проекций *ABT1* и *ABT2*;
- убедитесь в отсутствии ловушки связей;
- установите номер нормальной формы у отношений *ABT*, *ABT1* и *ABT2*.

Задание 2.51. Создайте средствами СУБД Access следующие отношения.

- *R1* – на основе бланка экзаменационной ведомости (первая страница), показанного на рис. 1.2;
- *R2* – на основе бланка экзаменационной ведомости (вторая страница), показанного на рис. 1.3.

Воспользуйтесь сведениями из табл. 1.2.

Имена реквизитов выберите самостоятельно. Приведите базу данных из отношений *R1* и *R2* к 3НФ. Создайте запросы с помощью SQL, формирующие полученные отношения в 3НФ из отношений *R1* и *R2*; обновляющие запросы для столбцов с числовыми значениями, позволяющие умножить эти значения на 2.

Задание 2.52. Дополните средствами СУБД Access построенные таким образом базы данных до нормального состава групп (15–20 студентов), предварительно построив необходимые входные формы для всех таблиц в ЗНФ.

Задание 2.53. Постройте запросы на SQL, позволяющие:

1) найти по заданным в условии запроса преподавателям фамилии студентов, сдающих им экзамены, с указанием даты и оценки;

2) рассчитать по заданным в условии запроса студентам их средний балл за сессию и указать, кто из них получит стипендию в следующем семестре (условием получения стипендии является средний балл студента, превышающий 4);

3) рассчитать средние баллы групп за сессию;

4) рассчитать количество различных оценок на экзамене в заданный день в заданной группе;

5) рассчитать количество неаттестованных студентов.

Задание 2.54. Постройте универсальное отношение на основе отношений в ЗНФ, полученных в задании 2.51, и убедитесь в отсутствии ловушки связей.

Задание 2.55. Проведите синтаксический анализ следующего SQL-запроса.

```
SELECT DISTINCTROW Водомость.[№_вед], Водомость.  
Код_пр, Водомость.Код_преп, Водомость.Код_гр, Водомость.  
ЧМГ, Строка.Код_ст, Строка.Оц, Студенты.ФИО, Студен-  
ты.Год_рожд, Преподаватели.ФИО_преп, Предметы.Назв_пр  
INTO Декомп_студ
```

```
FROM Студенты INNER JOIN (Преподаватели INNER JOIN  
(Предметы INNER JOIN (Водомость INNER JOIN Строка ON  
Водомость.[№_вед] = Строка.[№_вед]) ON Предметы.Код_предм  
= Водомость.Код_пр) ON Преподаватели.Код_преп = Водомость.  
Код_преп) ON Студенты.Код_студ = Строка.Код_ст;
```

и охарактеризуйте его результат. Воспользуйтесь данными табл. 1.2 и 1.3.

Задание 2.56. Рассмотрите отношение Поезда (см. табл. 1.5 и 1.6). Реализуйте его в базе данных средствами СУБД Access. Приведите отношение Поезда к ЗНФ. Реализуйте средствами SQL в той же базе данных набор проекций файла Поезда, полученных в результате перехода к ЗНФ.

Задание 2.57. Определите все пары отношений в 3НФ, из числа полученных в задании 2.56, которые допускают естественное соединение. Результаты таких соединений добавьте к множеству исходных отношений. Повторяйте поиск пар соединяемых отношений и расширение на этой основе множества отношений до тех пор, пока это возможно. Постройте граф соединений, вершинами которого являются полученные отношения. Дуга на графе между двумя вершинами проводится, если соответствующие отношения можно соединить. Создайте SQL-запросы для реализации всех соединений. Необходимые имена отношений выберите самостоятельно.

Задание 2.58. Определите все пары отношений в 3НФ, из числа полученных в задании 2.51, которые допускают естественное соединение. Результаты таких соединений добавьте к множеству исходных отношений. Повторяйте поиск пар соединяемых отношений и расширение на этой основе множества отношений до тех пор, пока это возможно. Постройте граф соединений, вершинами которого являются полученные отношения. Дуга на графе между двумя вершинами проводится, если соответствующие отношения можно соединить. Создайте SQL-запросы для реализации всех соединений. Необходимые имена отношений выберите самостоятельно.

Задание 2.59. Рассмотрите отношение со сведениями о научно-исследовательских работах. Список реквизитов отношения приведен ниже.

№ п/п	Наименование реквизита	Идентификатор
1	Название НИИ (научно-исследовательского института)	НИИ
2	Директор НИИ	Директор
3	Адрес НИИ	Адрес
4	Код отдела	Отдел
5	Число сотрудников в отделе	Ксотр
6	Код темы НИР (научно-исследовательской работы)	Тема
7	Дата начала темы	Датанач
8	Дата окончания темы	Датакон
9	Приоритет темы	Приор
10	Заказчик темы	Заказ
11	Объем финансирования темы заказчиком	Обфин
12	Код работы в теме	Работа
13	Продолжительность работы	Прод
14	ФИО исполнителя работы	ФИО

В отношении присутствует блок взаимно-однозначных соответствий для любой пары реквизитов из множества {НИИ, Директор, Адрес}. Далее реквизитом-представителем будем считать НИИ. Список функциональных зависимостей, в котором количество рассматриваемых вариантов сокращено по указанным выше принципам, приведен ниже.

Список функциональных зависимостей.

Отдел → НИИ, Отдел → Директор, Отдел → Ксотр, Тема → Датанач, Тема → Датакон, Тема → Приор, ФИО → НИИ, ФИО → Директор, ФИО → Отдел, Тема, Заказ → Обфин, Тема, Работа, ФИО → Прод.

Структура реляционной базы данных в ЗНФ

$R1(\text{НИИ}, \text{Директор}, \text{Адрес}),$

$R2(\text{НИИ}, \text{Отдел}, \text{Ксотр}),$

$R3(\text{Тема}, \text{Датанач}, \text{Датакон}, \text{Приор}),$

$R4(\text{ФИО}, \text{Отдел}),$

$R5(\text{Тема}, \text{Заказ}, \text{Обфин}),$

$R6(\text{Тема}, \text{Работа}, \text{ФИО}, \text{Прод}),$

$R7(\text{Тема}, \text{Заказ}, \text{Работа}, \text{ФИО}).$

Отношение $R7$ содержит первичный ключ для исходного множества реквизитов, а отношения $R1$ – $R6$ построены на основе зависимостей из минимального покрытия. Реализуйте отношения $R1$ – $R7$ средствами СУБД Access, заполните отношения согласованными между собой значениями и постройте SQL-запрос для вычисления универсального отношения.

Задание 2.60. Рассмотрим структуру реляционной базы данных в ЗНФ.

$R1(\text{НИИ}\#, \text{Директор}, \text{Адрес}),$

$R(\text{НИИ}, \text{Отдел}\#, \text{Ксотр}),$

$R3(\text{Тема}\#, \text{Датанач}, \text{Датакон}, \text{Приор}),$

$R4(\text{ФИО}\#, \text{Отдел}),$

$R5(\text{Тема}\#, \text{Заказ}\#, \text{Обфин}),$

$R(\text{Тема}\#, \text{Работа}\#, \text{ФИО}\#, \text{Прод}),$

$R(\text{Тема}\#, \text{Заказ}\#, \text{Работа}\#, \text{ФИО}\#).$

В схемах отношений реквизиты ключа отмечены знаком #. НИИ – сокращенное обозначение научно-исследовательского института.

Получить дерево соединений.

Ациклические базы данных

Методические указания

Ряд ограничений в предметной области и БД не может быть описан с помощью функциональных зависимостей, что привело к необходимости рассмотрения новых типов зависимостей – многозначных.

В отношении $R(A, B, C)$ имеется многозначная зависимость $A \twoheadrightarrow B$, если для любого a , являющегося значением реквизита A

$$im(BC)a = im(B)a \circ im(C)a,$$

где $im(B)a$ – множество значений реквизита B , связанных с заданным значением реквизита a ;

° – знак декартова произведения множеств.

Положение реквизитов B и C равноценно, поэтому одновременно справедливо $A \twoheadrightarrow C$, т.е. многозначные зависимости всегда встречаются парами.

Отношение $R(A, B, C)$ с многозначной зависимостью $A \twoheadrightarrow B$ содержит избыточную информацию, хотя и несколько другого рода, чем отношение в 1НФ. Оказывается, что отношения $R_1=R[A, B]$ и $R=R_2[A, C]$ вместе представляют всю информацию из R . Справедливо соотношение $R=R_1 * R_2$, которое можно считать равноценным определению многозначной зависимости.

Рассмотрим отношение $Z(\text{Завод}, \text{Изделие}, \text{Компл}, \text{План})$, где **Компл** – название комплектующего изделия для данного **Изделия**, а **План** – план выпуска Изделий. Справедлива функциональная зависимость **Завод, Изделие** \rightarrow **План** и мы имеем следующие отношения в 3НФ:

$Z_1(\text{Завод}, \text{Изделие}, \text{План})$,

$Z_2(\text{Завод}, \text{Изделие}, \text{Компл})$.

Отношение Z_1 содержит двухреквизитный ключ и в нем не может быть многозначной зависимости. В Z_2 существует МЗ вида **Изделие** \twoheadrightarrow **Завод** (и, следовательно, **Изделие** \twoheadrightarrow **Компл**), поскольку каждое изделие комплектуется одним и тем же набором комплектующих изделий, на каком бы заводе оно ни производилось. Отношение Z_2 необходимо разделить на два отношения $Z_{11}(\text{Завод}, \text{Изделие})$ и $Z_{12}(\text{Изделие}, \text{Компл})$. Вся информация из

Z_{11} содержится в Z_1 , поэтому окончательный список отношений состоит из Z_1 и Z_{12} .

Операция соединения позволяет получить $Z_2 = Z_{11} * Z_{12}$.

Известен специальный класс реляционных баз данных, названных ациклическими, для которых характерна однозначная декомпозиция на основе многозначных зависимостей и ряд других полезных (с точки зрения удобства обработки данных) свойств.

Для определения понятия «ациклическая схема базы данных» введем граф соединений на множестве отношений $\{S_1, S_2, \dots, S_k\}$. Вершинами графа соединений являются имена отношений S_1, S_2, \dots, S_k . Дуга графа $\langle S_i, S_j \rangle$ существует, если в структуре отношений S_i и S_j имеются общие реквизиты. Обозначим их через $A(i, j)$ и назовем весом дуги. Путь на графе соединений называется A -путем, если реквизит A содержится в структуре каждого отношения, лежащего на пути. В графе соединений требуется, чтобы для каждой пары отношений S_i, S_j с общим реквизитом $A(i, j)$ существовал $A(i, j)$ -путь между S_i и S_j . Если граф можно превратить в дерево с помощью исключения некоторых дуг при сохранении названного требования, то база данных с отношениями $\{S_1, S_2, \dots, S_k\}$ является ациклической.

Например, в графе соединений на рис. 2.1 можно разорвать любую из дуг и превратить граф в дерево. По этой причине база данных, состоящая из отношений S_1, S_2, S_3 , является ациклической.

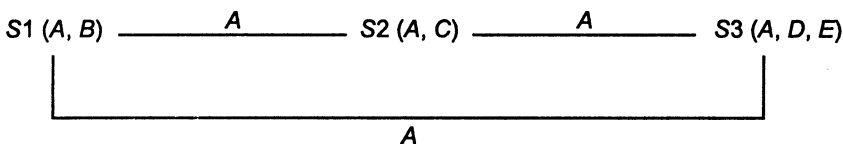


Рис. 2.1. Пример графа соединений

Алгоритм проверки структуры БД на ацикличность

Исходные данные – список отношений с указанием реквизитного состава каждого отношения.

Метод – последовательное выполнение указанных ниже шагов.

Шаг 1. Если некоторый реквизит встречается только в одном отношении, то необходимо вычеркнуть данный реквизит из этого отношения.

Шаг 2. Если все реквизиты некоторого отношения находятся среди реквизитов другого отношения, то первое отношение вычеркивается из списка.

Шаги 1 и 2 можно применять в любой последовательности.

Если в результате будут вычеркнуты все отношения, то база данных является ациклической. Если будут вычеркнуты не все отношения – база данных циклическая.

База данных с циклическим графом соединений может давать некорректные ответы на запросы из-за наличия нескольких неравноценных путей доступа при реализации запросов. В частности, отношение с реквизитами, полученными путем объединения весов дуг, образующих цикл, может быть вычислено несколькими способами и давать разные результаты. Для доказательства можно продемонстрировать наличие различающихся функциональных зависимостей в получаемых отношениях.

Рассмотрим простой пример графа соединений (рис. 2.2).

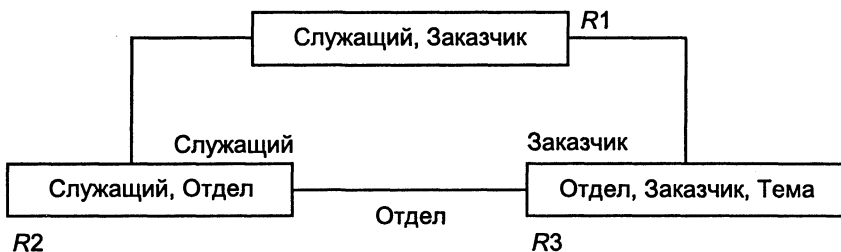


Рис. 2.2. Граф соединений ациклической базы данных

Наличие цикла очевидно. Отношение $T(\text{Служащий, Отдел, Заказчик})$ может быть вычислено либо как $T = R1 * R2$, либо в виде: $T = (R1 * R3)[\text{Служащий, Отдел, Заказчик}]$.

В первом случае будет справедлива функциональная зависимость $\text{Служащий} \rightarrow \text{Отдел}$, а во втором случае – нет, так как с заказчиками могут работать служащие, не являющиеся сотрудниками каких-то отделов.

Восстановление свойств ацикличности базы данных может быть проведено двумя способами.

1. Добавление в базу данных нового отношения с реквизитами, равными объединению весов дуг, образующих цикл. В этом

случае придется допустить существование неопределенных значений в новом отношении.

2. Добавление новых реквизитов, переименование и разделение ролей реквизитов. Такое решение не требует дополнительных соглашений при интерпретации запросов и не создает дополнительных неопределенных значений. Приведем типичные варианты разделения и переименования реквизитов.

- При структуре функциональных зависимостей вида $A \rightarrow B \rightarrow C$, $A \rightarrow E \rightarrow C$ необходимо установить две разные роли для реквизита C (например, $C1$ и $C2$). Рассмотрим БД с реквизитами Студент, Группа, Куратор, Кафедра и зависимостями Студент \rightarrow Группа \rightarrow Кафедра, Студент \rightarrow Куратор \rightarrow Кафедра. В первом случае подразумевается скорее всего выпускающая кафедра для группы студентов, а во втором – кафедра, на которой работает куратор. После разделения реквизита Кафедра получаем структуру БД в виде:

$P1(\text{Студент, Группа}),$

$P2(\text{Группа, Выпускающая_кафедра}),$

$P3(\text{Студент, Куратор}),$

$P4(\text{Куратор, Кафедра_куратора}).$

- Каждое применение теоремы Т6 свидетельствует о цикличности базы данных. Рассмотрим, например, зависимости Служащий \rightarrow Отдел и Отдел, Заказчик \rightarrow Тема. Эта ситуация отражена на рис. 2.2. Для преодоления цикличности необходимо разделить роли реквизита Отдел, например, ввести реквизит Отдел_служащего.

- Структура функциональных зависимостей вида $AB \rightarrow C$, $C \rightarrow B$ обычно возникает, если действия, совершаемые объектом, приписываются классу объектов. В таком случае необходимо добавить в структуру базы данных реквизит, обозначающий этот объект. Например, служащий фирмы покупает билет в аэропорту и счет направляется в отделение фирмы. Поэтому справедливы зависимости Аэропорт, Фирма \rightarrow Отделение и Отделение \rightarrow Фирма. Добавив реквизит Служащий, мы разрушаем нежелательную структуру функциональных зависимостей и получаем – Аэропорт, Служащий \rightarrow Отделение и Отделение \rightarrow Фирма.

Следует отметить, что циклическая база данных может содержать два, три и более циклов. Описанный выше алгоритм фиксирует наличие циклов вообще. Поэтому когда циклическая база данных преобразована рекомендуемыми здесь способами, необходимо заново проверить ее на ацикличность.

Учитывая высокую трудоемкость приведения отношений к 3НФ при достаточно большом числе реквизитов (в качестве границы назовем 1–14 реквизитов), можно рекомендовать следующие действия при проектировании структуры реляционной БД:

- каждый входной документ привести к 3НФ и установить первичный ключ в каждом случае;

- для полученного на шаге 1 множества отношений построить граф соединений. Если граф соединений можно преобразовать в дерево соединений (или алгоритм проверки ацикличности дал положительный результат), то база данных в целом является ациклической и соответствует 3НФ. В противном случае для достижения ацикличности необходимо выполнить преобразования, рекомендованные выше.

Критерии, которым соответствует база данных в 3НФ и ациклическая база данных, безусловно не совпадают. В первую очередь, ациклическая база данных не гарантирует минимальную избыточность представления информации. Гарантии единственного пути доступа в ациклической базе данных, вероятно следует признать более существенными для пользователей-непрофессионалов. Надо также учитывать элементарность метода проверки ацикличности базы данных в сравнении с необходимостью формального анализа функциональных зависимостей, требуемых при создании базы данных в 3НФ.

Задания

Задание 2.61. Выполните декомпозиции и соединения с отношениями

Z1(Завод, Изделие, План),

Z2(Завод, Изделие, Компл)

из методических указаний к пункту 2.4 средствами СУБД. Значения в отношениях выберите таким образом, чтобы соблюдалась многозначная зависимость Изделие $\rightarrow\rightarrow$ Компл. Убедитесь, что после декомпозиции избыточность данных сократилась.

Задание 2.62. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Служащий, Организация, Должность).
R2(Организация, Расчетный_счет, Адрес).
R3(Служащий, Автомобиль).
R4(Автомобиль, Цена, Регистрационный_номер).

Задание 2.63. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Деталь, Станок).
R2(Деталь, Производитель).
R3(Производитель, Станок, Количество).
R4(Станок, Вес, Год_выпуска).

Задание 2.64. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(ФИО, Подразделение).
R2(ФИО, Должность).
R3(Подразделение, Должность, Зарплата).
R4(Должность, Вид_работы).
R5(Подразделение, Номер).

Задание 2.65. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Товар, Поставщик).
R2(Поставщик, Склад).
R3(Товар, Магазин).
R4(Магазин, Склад, Адрес_магазина).

Задание 2.66. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(СУБД, Тип_подключения, Размер).
R2(Тип_подключения, Сетевой_протокол, Клиент).
R3(СУБД, Фирма, Срок_использования).
R4(Фирма, Клиент, Срок_контракта).
R5(Клиент, Телефон).

Задание 2.67. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Статья, Автор, Тема).

R2(Журнал, Тема).

R3(Раздел, Количество_статей, Количество_авторов).

R4(Автор, Должность).

Задание 2.68. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Студент, Секция, День_посещения).

R2(Секция, Вид_спорта, Преподаватель).

R3(Студент, Группа, Дисциплина).

R4(Дисциплина, Преподаватель).

Задание 2.69. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Кинотеатр, Фильм).

R2(Кинотеатр, Сеанс).

R3(Фильм, Сеанс, Стоимость_билета).

R4(Сеанс, Время).

R5(Фильм, Продолжительность).

Задание 2.70. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Кафедра, Студент, Специализация).

R2(Кафедра, Преподаватель).

R3(Студент, Группа, Успеваемость).

R4(Преподаватель, Телефон).

R5(Студент, Дисциплина, Преподаватель).

Задание 2.71. Определите первичные ключи в каждом отношении. Установите, является ли база данных в целом ациклической. Если база данных циклическая, то приведите ее к ациклическому виду.

R1(Программист, Страна, Возраст).
R2(Страна, Язык).
R3(Программист, Проект, Страна).
R4(Проект, Язык_программирования).

2.5.

Сетевая и иерархическая модели данных

Методические указания

Информационными конструкциями в сетевой модели данных являются отношения и веерные отношения. Понятие «отношение» уже рассматривалось применительно к реляционной модели данных и будет использоваться здесь без изменений, хотя в некоторых сетевых СУБД допускаются отношения с многоуровневой (три и более) структурой.

Сетевая БД представляется как множество отношений и веерных отношений. Отношения разделяются на основные и зависимые.

Веерным отношением $W(R, S)$ называются два отношения (одно основное отношение R и одно зависимое отношение S) и связь между ними при условии, что каждое значение зависимого отношения связано с единственным значением основного отношения. Названное условие является ограничением, характерным для сетевой модели данных в целом. Способ реализации этого ограничения в памяти компьютера неодинаков у различных сетевых СУБД.

Допустимые в сетевой модели данных операции представляют собой различные варианты выборки.

Сетевые базы данных в зависимости от ограничений на вхождение отношений в веерные отношения разделяются на много- и двухуровневые сети.

Ограничение двухуровневых сетей состоит в том, что каждое отношение может существовать в одной из перечисленных ниже ролей:

- вне каких-либо веерных отношений;
- в качестве основного отношения в любом количестве веерных отношений;
- в качестве зависимого отношения в любом количестве веерных отношений.

Запрещается использование отношения в качестве основного в одном контексте и одновременно в качестве зависимого – в другом контексте.

Многоуровневые сети не предусматривают никаких ограничений на взаимосвязь верных отношений, в некоторых сетевых СУБД разрешены даже циклические структуры сети.

Среди известных в настоящее время сетевых СУБД наиболее распространены системы, поддерживающие двухуровневую сеть. Операция связывания отношений в реляционных СУБД также приводит к двухуровневым системам отношений. Двухуровневые сети обладают свойством ацикличности.

Для двухуровневых сетевых СУБД вводятся еще два ограничения (с теоретической точки зрения необязательные):

- первичный ключ основного отношения может быть только одноквалификативным;
- верное отношение устанавливается, если первичный ключ основного отношения является частью первичного ключа зависимого отношения.

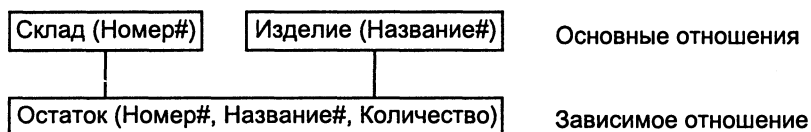
В структуру основного и зависимого отношений вводится дополнительный реквизит, называемый адресом связи. Значения адресов связи совместно обеспечивают в верном отношении соответствие каждого значения зависимого отношения S с единственным значением основного отношения R .

Значение отношения при хранении в памяти компьютера часто называется записью. Адресом связи называется реквизит в составе записи, в котором хранится начальный адрес или номер следующей обрабатываемой записи.

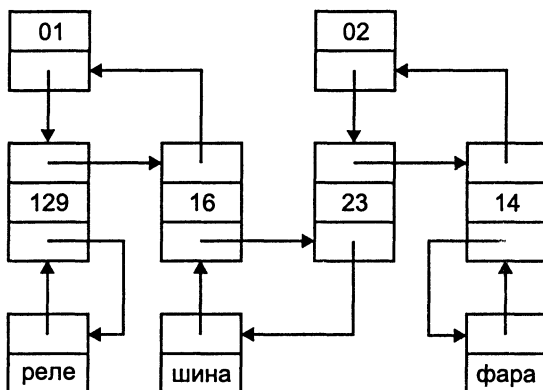
Связь значений зависимого отношения с единственным значением основного отношения в простейшем случае обеспечивается следующим образом. Адрес связи некоторой записи основного отношения указывает на одну из записей зависимого отношения (значением адреса связи основного отношения является начальный адрес этой записи зависимого отношения), адрес связи указанной записи зависимого отношения – на следующую запись зависимого отношения, связанную с той же записью основного отношения и т.д. Последняя запись зависимого отношения в данной цепочке адресует названную выше запись основного отношения.

Таким образом, получается кольцевая структура адресов связи, называемая веером, где роль «ручки» веера играет запись ос-

нового отношения. На графических иллюстрациях сетевых баз данных адрес связи изображается стрелкой, направленной от адреса связи данной записи к той записи, чей начальный адрес (номер) служит значением этого адреса связи. На рис. 2.3 показаны структуры и значения веерных отношений двух простых сетевых двухуровневых баз данных. Реквизиты первичного ключа во всех случаях помечены знаком #.



Значения основного значения Склад



Значения основного отношения Изделие

Рис. 2.3. Сетевая база данных «Склад–Изделие»

Из аналогии определений веерного отношения и функциональной зависимости следует утверждение: если существует веерное отношение, то ключ зависимого отношения функционально определяет ключ основного отношения и наоборот, если ключ одного отношения функционально определяет ключ второго отношения, то первое отношение может быть зависимым, а второе – основным в некотором веерном отношении.

Указанный факт обычно используется для того, чтобы при наличии функциональной зависимости между первичными ключами двух отношений доказать корректность связывания этих отношений в веерное отношение.

В схеме сетевой БД отношения и веерные отношения часто трактуются как файлы и связи, что позволяет рассматривать сетевую структуру как множество файлов

$$F = \{F1(X1), F2(X2), \dots, Fi(Xi), \dots, Fn(Xn)\},$$

где Xi – реквизиты ключа в файле Fi .

Дополнительно вводится граф сетевой структуры B с вершинами $\{X1, X2, \dots, Xi, \dots, Xn\}$. Дуга $\langle Xi, Xj \rangle$ в графе B означает, что Xi является частью Xj и $Fj[Xi]$ является подмножеством Fi . Последнее условие имеет тот же смысл, что и синтаксическое включение отношений в реляционной модели данных. Здесь предполагается, что ключ основного файла содержится в зависимом файле. Граф B аналогичен графу соединений для реляционной БД.

Сетевая база данных DBA называется ациклической, если между любыми двумя вершинами на графе B существует не более одного пути. Двухуровневые сети всегда ациклические.

Для множества файлов F ациклической базы данных DBA вполне применима операция

$$m(DBA) = F1 \& F \& \dots \& Fi \& \dots \& Fn,$$

называемая максимальным пересечением. Ее аналогом может служить последовательность соединений в реляционной базе данных.

Рассмотрим алгоритм формирования структуры двухуровневой сетевой базы данных на основе известного множества реквизитов и функциональных зависимостей.

Исходное множество функциональных зависимостей и реквизиты первичного ключа получаются так же, как при формировании множества отношений в 3НФ.

Алгоритм поясняется на примере базы данных о научно-исследовательских работах. Список реквизитов приведен ниже.

Наименование реквизита	Идентификатор
Название НИИ	НИИ
Директор НИИ	Директор
Адрес НИИ	Адрес
Код отдела	Отдел
Число сотрудников в отделе	Ксотр
Код темы НИР	Тема
Дата начала темы	Датанач
Дата окончания темы	Датакон
Приоритет темы	Приор
Заказчик темы	Заказ
Объем финансирования темы заказчиком	Обфин
Код работы в теме	Работа
Продолжительность работы	Прод
ФИО исполнителя работы	ФИО

Алгоритм получения двухуровневой структуры сети

Исходные данные – список реквизитов и функциональных зависимостей в базе данных.

Шаг 1. Для каждой функциональной зависимости вида $A \rightarrow B$ создается файл $F_i(A, B)$. Каждый блок взаимно-однозначных соответствий также порождает файл с ключом, равным старшему по объему понятия реквизиту.

В нашем примере будут созданы следующие файлы (ключи помечены знаком #):

- $F_1(\text{НИИ \#, Директор, Адрес}),$
- $F_2(\text{Отдел \#, НИИ, Ксотр}),$
- $F_3(\text{Тема \#, Датанач, Датакон, Приор}),$
- $F_4(\text{ФИО \#, Отдел}),$
- $F_5(\text{Тема \#, Работа \#, ФИО \#, Прод}),$
- $F_6(\text{Тема \#, Заказ \#, Обфин}).$

Шаг 2. У всех пар файлов, полученных на шаге 1, проверяется условие для ключей (K_i является частью K_j). Если оно соблюдается, то из соответствующих файлов создается веерное отношение $W_{ij}(F_i, F_j)$.

В нашем примере получим

$W_{35}(F_3, F_5), W_{45}(F_4, F_5), W_{36}(F_3, F_6).$

Шаг 3. Если на шаге 2 будут получены два верных отношения Wij и Wjk , то все реквизиты файла Fi передаются в файл Fj и Fi вместе с Wij уничтожаются.

В нашем примере таких верных отношений нет.

Шаг 4. Реквизиты, не вошедшие в состав верных отношений на шаге 2, добавляются в те файлы Fn (точнее, в содержащие Fn верные отношения), где они будут неключевыми. При наличии нескольких подходящих файлов предпочтение отдается основным файлам. Если требуемые Fn отсутствуют, то создается новый файл из реквизитов первичного ключа, и повторяются шаги 2, 3, 4.

В нашем примере $F4$ расширяется реквизитами НИИ, Директор, Адрес, Ксотр.

На рис. 2.4 показана структура соответствующей двухуровневой сетевой базы данных.

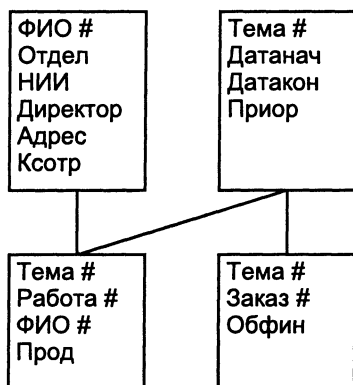


Рис. 2.4. Информационная база НИИ (Тема)

Структуры основных отношений показаны в верхней части рис. 2.4, а структуры зависимых отношений – в нижней части.

Перед рассмотрением операций в сетевой базе данных следует отметить, что существуют два разных подхода к обработке данных средствами СУБД.

1. Применение базового языка программирования. При этом подходе для манипулирования данными в СУБД разрабатывается универсальный язык программирования, обеспечивающий обращения к базе данных, работу с переменными и остальные возможности.

2. Применение включающего языка программирования. Включающий язык – это обычный язык программирования (например, Паскаль), в котором обращения к базе данных реализуются с помощью подпрограмм. Среди параметров, передаваемых подпрограмме, указываются название операции, имена обрабатываемых отношений и др. Включающий язык используется практически во всех известных сетевых СУБД. Это объясняется принципом доступа к данным в сетевой базе данных, который может быть назван навигационным.

Центральным для навигационного принципа доступа является понятие «текущая запись» в отношениях базы данных. Текущей записью в отношениях после выполнения некоторой операции является значение отношения, на котором операция завершилась. Следующая операция начинается с этой текущей записи, а в результате выполнения операции положение текущей записи изменяется (завершение операции может изменить положение текущей записи и в других отношениях).

Рассмотрим операции выборки для двухуровневой сетевой базы данных. Для того чтобы не пользоваться синтаксисом включающего языка, условимся записывать лишь название операции и условие выборки. Примеры выборки относятся к сетевой структуре, изображенной на рис. 2.5.

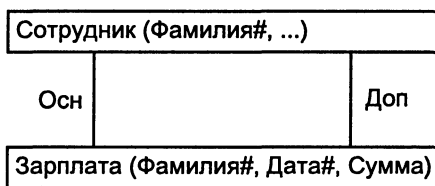


Рис. 2.5. Сетевая база данных «Сотрудник–Зарплата»

В этой базе данных на основном отношении Сотрудник и зависимом – Зарплата установлено два веерных отношения: Осн – основная зарплата и Доп – дополнительная зарплата.

1. Операция OBTNM – получить запись в основном отношении.

OBTNM (Сотрудник = «Иванов»).

После выполнения указанной выборки в отношении Сотрудник в качестве текущей будет установлена запись с значением пер-

вичного ключа «Иванов». Затем реквизиты этой текущей записи могут обрабатываться средствами включающего языка (они становятся значениями переменных, печатаются и т.п.).

2. Операция OBTNF – получить запись в зависимом отношении.

OBTNF(Сотрудник = «Иванов», Зарпл*Осн).

При выборке в зависимом отношении текущей записью становится следующая запись зависимого отношения относительно той, которая раньше была текущей в зависимом отношении. Условие выборки содержит указание на текущую запись в основном отношении, а также на имя зависимого отношения и имя верного отношения. В результате выполнения двух операторов

OBTNM (Сотрудник = «Иванов»);

OBTNF(Сотрудник = «Иванов», Зарпл*Осн).

при условии, что обращения к отношению Зарпл ранее не проводились, будут получены сведения о первой (с момента поступления на работу) основной зарплате Иванова.

Средствами включающего языка может быть организован циклический процесс, и тогда возможны более сложные варианты доступа. Рассмотрим, например, последовательность операций

OBTNM (Сотрудник = «Иванов»)

M1: OBTNF(Сотрудник = «Иванов», Зарпл*Осн)

print Зарпл

goto M1.

Оператор print печатает все реквизиты текущей записи отношения Зарпл. На первый взгляд, использование безусловного перехода создает заикливание при выполнении. Однако операторы выборки в сетевых СУБД по окончании выборки вырабатывают код возврата, и выход за последнюю запись в отношении Зарпл сопровождается специальным значением этого кода в команде OBTNF, означающим неудачное завершение выборки. Значение кода возврата всегда проверяется средствами включающего языка и этим обеспечивается выход из цикла. Приведенная выше последовательность операций напечатает все сведения о получении Ивановым основной заработной платы.

Практически в сетевых СУБД количество операций выборки достаточно велико. Мы рассмотрели минимально необходимое множество вариантов выборки. Остальные варианты выборки создают более удобные для прикладного программиста возможности реализации запросов.

Иерархическая модель данных имеет много общих черт с сетевой моделью данных, хронологически она появилась даже раньше, чем сетевая. Допустимыми информационными конструкциями в иерархической модели данных являются отношение, веерное отношение и иерархическая база данных. В отличие от рассмотренных ранее моделей данных, где предполагалось, что информационным отображением одной предметной области является одна база данных, в иерархической модели данных допускается отображение одной предметной области в нескольких иерархических базах данных.

Понятия «отношение» и «веерное отношение» в иерархической модели данных не изменяются.

Иерархической базой данных называется множество отношений и веерных отношений, для которых соблюдаются два ограничения.

1. Существует единственное отношение, называемое корневым, которое не является зависимым ни в одном веерном отношении.

2. Все остальные отношения (за исключением корневого) являются зависимыми отношениями только в одном веерном отношении.

Схема иерархической базы данных по составу компонент идентична сетевой базе данных. Перечисленные выше ограничения поддерживаются иерархическими СУБД.

На рис. 2.6 изображена структура иерархической базы данных, представляющая студентов и преподавателей вуза, и удовлетворяющая всем ограничениям, указанным в определении.

В графических иллюстрациях структуры иерархической базы данных приводятся названия соответствующих отношений.

Ограничение, которое поддерживается в иерархической модели данных, состоит в невозможности нарушения требований, фигурирующих в определении иерархической базы данных. Это ограничение обеспечивается специальной укладкой значений отношений в памяти компьютера. Ниже будет рассмотрена одна из простейших реализаций укладки иерархической базы данных.

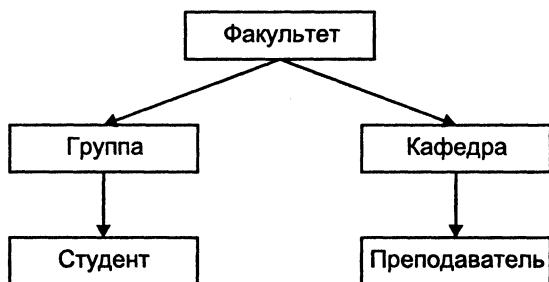
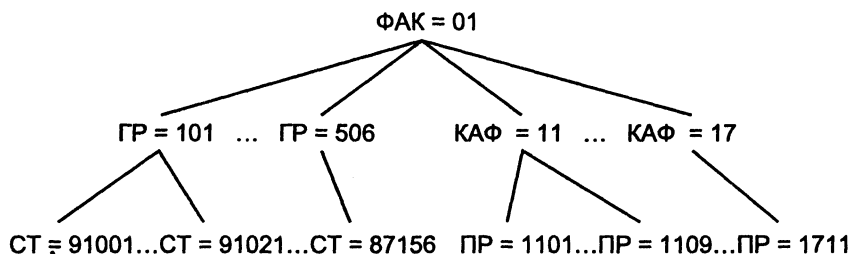


Рис. 2.6. Иерархическая база данных «Вуз–Кафедра»



а

Запись 1

ФАК = 01	ГР = 101	СТ = 91001	СТ = 91002	...	СТ = 91021	ГР = 102	СТ = 91022
...	ГР = 506	...	СТ = 87156	КАФ = 11	ПР = 1101	ПР = 1102	...
КАФ = 12	ПР = 1201	...	КАФ = 17	...	ПР = 1711		

Запись 2

ФАК = 02	ГР = 110	СТ = 91188	...	СТ = 91021	ГР = 111	...
----------	----------	------------	-----	------------	----------	-----

б

Рис. 2.7. Экземпляр иерархической базы данных «Вуз–Кафедра»

На рис. 2.7 а приведена связь значений отношений из иерархической базы данных, структура которой показана на рис. 2.6. Каждое значение представляется соответствующей величиной первичного ключа. Использованы следующие сокращения: ФАК –

факультет, ГР – группа, КАФ – кафедра, СТ – студент, ПР – преподаватель.

Далее эта информация организуется в линейную последовательность значений (рис. 2.7 б).

Необходимо отметить, что известны различные возможности прохождения иерархически организованных значений в линейной последовательности. Принцип, применяемый для иерархических баз данных, называется концевым прохождением.

Правила концевого прохождения следующие:

Шаг 1. Начиная с первого значения корневого отношения перечисляются первые значения соответствующих отношений на каждом уровне вплоть до последнего.

Шаг 2. Перечисляются все значения в том веерном отношении, на котором остановился шаг 1.

Шаг 3. Перечисляются значения всех вееров этого веерного отношения.

Шаг 4. От достигнутого уровня происходит подъем на предыдущий уровень, и если возможно применить шаг 1, то процесс повторяется.

Записью иерархической базы данных называется множество значений, содержащих одно значение корневого отношения и все вееры, доступные от этого значения в соответствии со структурой иерархической базы данных. В нашем примере одну запись образуют данные, относящиеся к одному факультету.

Для веерных отношений в составе иерархической базы данных справедлива уже известная закономерность: если существует веерное отношение, то ключ зависимого отношения функционально определяет ключ основного отношения, и наоборот, если ключ одного отношения функционально определяет ключ второго отношения, то первое отношение может быть зависимым, а второе – основным в некотором веерном отношении. Кроме того, ограничение на наличие единственного корневого отношения в иерархической базе данных трансформируется в требование: первичный ключ каждого некорневого отношения должен функционально определять первичный ключ корневого отношения.

Рассмотрим алгоритм формирования иерархической БД на основе известного множества реквизитов и функциональных зависимостей. Исходное множество функциональных зависимостей и реквизиты первичного ключа получаются так же, как при фор-

мировании множества отношений в 3НФ. Алгоритм иллюстрируется тем же примером, что и для двухуровневой сети.

Алгоритм получения структуры иерархической БД

Исходные данные – список реквизитов и функциональных зависимостей в базе данных.

Шаг 1. Для каждой функциональной зависимости вида $A \rightarrow B$ создать отношение $Si(A, B)$. Каждый блок взаимно-однозначных соответствий также порождает отношение с ключом, равным старшему по объему понятия реквизиту.

В нашем примере будут созданы следующие отношения (ключи помечены знаком #):

$S1(\text{НИИ \#, Директор, Адрес}),$
 $S2(\text{Отдел \#, НИИ, Ксотр}),$
 $S3(\text{Тема \#, Датанач, Датакон, Приор}),$
 $S4(\text{ФИО \#, Отдел}),$
 $S5(\text{Тема \#, Работа \#, ФИО \#, Прод}),$
 $S6(\text{Тема \#, Заказ \#, Обфин}).$

Шаг 2. Разделить отношения на группы по признаку: два отношения находятся в общей группе, если их ключи функционально определяют хотя бы один общий реквизит.

Для отношений $S1 - S6$ получаем две группы:

- $S1, S2, S4, S5$ (все ключи функционально определяют реквизит НИИ);
- $S3, S6, S5$ (все ключи функционально определяют реквизит ТЕМА).

Далее шаги 3, 4, 5 выполняются отдельно для каждой группы. Количество групп определяет количество иерархических БД.

Шаг 3. У всех пар отношений группы проверить условие для ключей отношений $K_j \rightarrow K_i$. Если оно соблюдается, то из соответствующих отношений создается веерное отношение $W_{ij}(S_i, S_j)$.

Шаг 4. Найти в группе цепи веерных отношений и образовать из них дерево. Элемент цепи образуется по условию $W_{ij} - W_{jk}$.

В нашем примере получим:

группа 1) – цепь $W12(S1, S2), W24(S2, S4), W45(S4, S5)$ образует дерево;

группа 2) – цепей нет, но $W35(S3, S5)$ и $W36(S3, S6)$ образуют дерево.

Шаг 5. Реквизиты, оставшиеся вне цепей на шаге 4, добавить в структуру тех отношений, где они будут неключевыми, либо в структуру отношений, соответствующих висячим вершинам дерева.

Шаг 6. Если группы, полученные на шаге 2, содержат общие отношения, то необходимо решить вопрос о целесообразности установления логических связей между иерархическими БД.

Шаг 7. Сократить список реквизитов в сегментах за счет удаления реквизитов зависимого отношения, общих в паре «основной – зависимый».

Итоговая иерархическая структура для рассматриваемого примера показана на рис 2.8. Она содержит две иерархические базы данных. В некоторых иерархических СУБД не допускается логическая связь баз данных, так как формально это является нарушением ограничения иерархической модели данных.

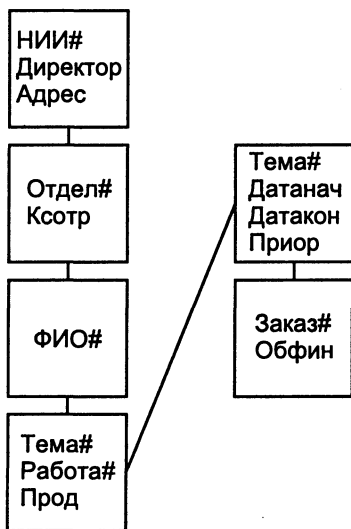


Рис. 2.8. Иерархическая база данных НИИ

Манипулирование иерархической базой данных происходит с применением включающего языка. Подпрограмма обращения к базе данных содержит ряд параметров, из которых мы будем использовать лишь код требуемой операции и одно или несколько условий выбора. По причинам, которые уже были перечисле-

ны при рассмотрении сетевой модели данных, для иерархической модели данных необходимы только операции выборки.

Минимальное множество вариантов выборки соответствует трем операциям:

1. GU – получить уникальную запись по известным значениям первичного ключа на каждом уровне дерева иерархической базы данных.

2. GN – получить следующую запись на том уровне дерева, где находится текущая запись после выполнения оператора GU.

3. GNP – получить следующую запись на расположенном непосредственно ниже уровне дерева, относительно позиции, где находится текущая запись после выполнения оператора GU или GN.

Например, для запроса «получить информацию о преподавателе с кодом 1103 кафедры 11 факультета 01» потребуется оператор

```
GU Факультет (Факультет = "01")  
  Кафедра (Кафедра = "11")  
    Преподаватель (Преподаватель = "1103")  
print Преподаватель.
```

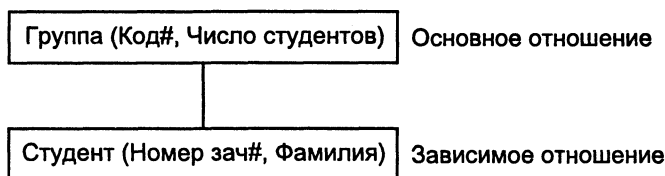
В этом примере названия отношений совпадают с названиями соответствующих первичных ключей.

Запрос «получить список всех студентов группы 10» реализуется следующей последовательностью операторов

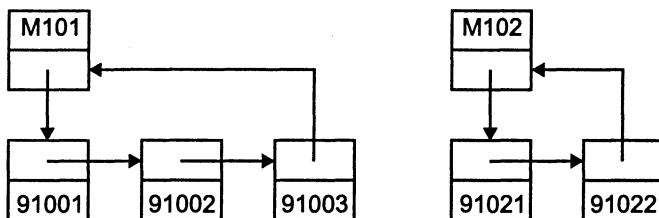
```
GU Факультет (Факультет = "01")  
  Группа (Группа="10")  
    Студент  
M1: GN Студент  
print Студент  
goto M1.
```

Поскольку в операторе GU не указано условие выборки в отношении Студент, текущей записью станет первая запись этого отношения и далее циклическое повторение оператора GN обеспечит требуемое извлечение всех записей о студентах группы 10. Выход из цикла произойдет в результате получения кода возврата «конец отношения» и эта ситуация должна проверяться средствами включающего языка.

Задание 2.72. Реализуйте изображенную ниже сетевую структуру (рис. 2.9) средствами СУБД Access.



Значения основного отношения Группа



Значения зависимого отношения Студент

Рис. 2.9. Экземпляр двухуровневой сети «Студент–Группа»

Для установления адресных цепочек используйте дополнительный реквизит Адрес_связи, значениями которого будут номера записей. Доведите количество студентов до 15–20 в каждой группе.

Задание 2.73. Реализуйте изображенную на рис. 2.5 сетевую структуру «Сотрудник – Зарплата» средствами СУБД Access.

Для установления адресных цепочек используйте дополнительный реквизит Адрес_связи, значениями которого будут номера записей. Доведите количество сотрудников до 5–10, и интервал получения зарплат – до года.

Задание 2.74. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
ФИО студента	W(ФИО, Группа)
Группа	S(Преподаватель, Кафедра)
Дисциплина	Z(ФИО, Дисциплина, Преподаватель, Оценка)
Преподаватель	Y(Группа, Преподаватель, Дисциплина, День)
Кафедра	F(Дисциплина, Кафедра)
Оценка экзамена	
День занятий	

Студенты и преподаватели-однофамильцы отсутствуют.

Задание 2.75. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
ФИО сотрудника	W(фио, код отдела, код учреждения)
код отдела	S(тема, оборудование, фио)
тема работы	T(тема, заказчик, объем финансирования)
Оборудование	Y(тема, дата начала, дата окончания)
дата начала темы	
дата окончания темы	
код учреждения	
объем финансирования темы	
Заказчик	

Тему могут финансировать несколько заказчиков. Сотрудников-однофамильцев нет.

Задание 2.76. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
ФИО сотрудника	W(фио, предприятие)
код предприятия	S(объект, оборудование, поставщик, дата начала поставки)
код объекта	Z(фио, объект)
код оборудования	Y(объект, адрес строительства)
код поставщика	T(поставщик, адрес поставщика)
дата начала поставки	
адрес строительства	
адрес поставщика	

Сотрудников-однофамильцев нет. Сотрудник может участвовать в строительстве нескольких объектов.

Задание 2.77. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
Команда	W (команда, главный тренер)
главный тренер	S (команда, игрок, число выступлений)
Игрок	T (команда, команда-гость, дата, счет)
число выступлений	Y (команда, дата, число очков)
дата игры	
команда-гость	
счет игры	
число очков	

Игроки-однофамильцы отсутствуют. Переход игроков в другие команды разрешен.

Задание 2.78. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
код поставщика	W (изделие, цена)
код покупателя	S (поставщик, покупатель, изделие, количество)
код изделия	T (покупатель, адрес)
цена изделия	R (покупатель, склад, изделие, количество)
Количество изделий	
адрес покупателя	
код склада	

Количество изделий в S и количество изделий в R представляют собой разные реквизиты.

Задание 2.79. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
код предприятия	W (предприятие, город)
Город	S (предприятие, изделие, год, фактический выпуск)
код изделия	T (предприятие, номер контейнера, дата, станция назначения)
Год	Y (номер контейнера, емкость контейнера)
Фактический выпуск	
номер контейнера	
Емкость контейнера	
дата отправки (дата)	
Станция назначения	

Данные приводятся по одному городу.

Задание 2.80. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
ФИО сотрудника (фio)	W (фio, дата рождения, образование)
дата рождения	S (фio, место работы, должность, дата начала работы)
Образование	T (место работы, должность, минимальная зарплата)
место работы	Y (место работы, директор)
Должность	
дата начала работы	
Минимальная зарплата	
Директор	

Сотрудников-однофамильцев нет. В отношении S перечисляются все места работы сотрудников.

Задание 2.81. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
Издательство	W (автор, адрес)
Город	S (издательство, город)
Название книги	T (издательство, автор, название книги)
Автор	R (название книги, цена)
адрес автора	Y (номер магазина, дата, город, название книги, сумма продажи)
Цена	
номер магазина	
Дата	
сумма продажи	

Авторов-однофамильцев нет, названия издательств не повторяются, книги данного названия имеют одинаковую цену.

Задание 2.82. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
Табельный № рабочего	R (табельный №, дата, код детали, количество)
фio рабочего (фio)	S (склад, дата, код детали, количество)
код детали	T (изделие, код детали, количество)
Склад	W (табельный №, фio)
Изделие	Y (код детали, цена детали)
Дата	
цена детали	
Количество	

Количество деталей в S , количество деталей в R и количество деталей в T представляют собой разные реквизиты.

Задание 2.83. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
Музей	W (музей, город)
Город	S (музей, экспонат, год поступления)
Экспонат	T (экспонат, фио, год реставрации)
Год	Z (экспонат, автор)
фио реставратора	
Автор	

Названия музеев и экспонатов не повторяются.

Задание 2.84. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
Наименование судна	W (судно, дата начала рейса, порт прибытия)
Грузоподъемность судна	S (судно, код страны, грузоподъемность, код судовладельца)
дата начала рейса	Y (порт, код страны)
код судовладельца	Z (судно, порт ремонта).
Порт	
код страны	

Порт означает порт прибытия или порт ремонта.

Задание 2.85. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
отделение банка	W (отделение, № счета, остаток средств)
Клиент	S (клиент, № счета, адрес)
№ счета	E (отделение, ревизор, № счета, дата ревизии)
адрес клиента	Y (отделение, план прибыли)
остаток средств	
Ревизор	
дата ревизии	
план прибыли текущего года	

Некоторые номера счетов открыты в нескольких отделениях банка.

Задание 2.86. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
Дата	R (табельный №, фио)
Цех	S (дата, участок, табельный №, код причины, время простоя)
Участок	Y (цех, участок, фио мастера участка)
Табельный № рабочего фио рабочего (фио) код причины простоя процент оплаты время простоя фио мастера участка	T (код причины простоя, процент оплаты)

В период простоя рабочий может работать на другом участке.

Задание 2.87. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
код материала	R (цех, код изделия, план года)
цена материала	F (код материала, цена материала)
код изделия	G (код изделия, цена изделия)
цена изделия	T (код материала, код изделия, норма расхода)
цех	
план текущего года	
норма расхода материала	

Задание 2.88. Разработайте структуру двухуровневой сетевой базы данных для заданного множества реквизитов и отношений.

Реквизиты	Отношения
пациент	W (пациент, домашний адрес)
домашний адрес	S (№ больницы, пациент, врач, № палаты)
номер больницы	T (пациент, врач, лекарство)
врач	Z (врач, № больницы)
номер палаты	L (лекарство, норма запаса)
лекарство	
норма запаса	

Информация хранится для одной больницы.

Задание 2.89. Для приведенной ниже иерархической структуры базы данных укажите минимально возможный набор реквизитов в отношениях.

Реквизиты: Музей, Город, Экспонат, Год, Выставка, ФИО реставратора.

Отношения: $W(\text{Музей, Город})$, $C(\text{Экспонат, Год поступления})$, $T(\text{Экспонат, Год реставрации, ФИО})$, $S(\text{Выставка, Экспонат, Год выставки})$.

Всерные отношения: (W, C) , (C, S) , (C, T) .

Названия музеев и выставок не повторяются.

Задание 2.90. Детализируйте приведенную ниже иерархическую структуру (рис. 2.10). Необходимые имена реквизитов выбрать самостоятельно. Проверьте соблюдение всех требований алгоритма получения структуры иерархической базы данных.

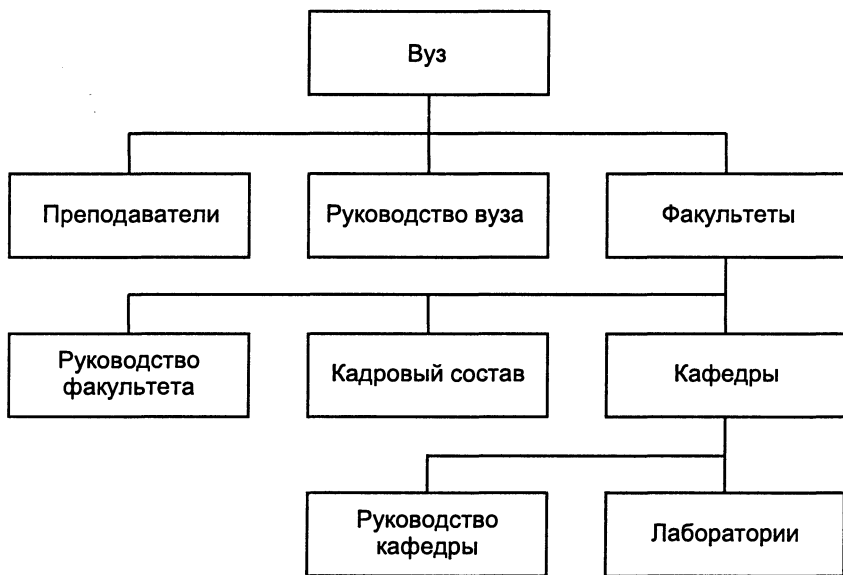


Рис. 2.10. Структура вуза

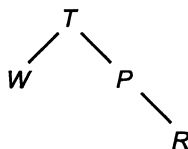
Задание 2.91. Реализуйте иерархическую структуру. Разрешается добавлять или исключать имена реквизитов в отношениях. Разработайте пример записи иерархической базы данных.

Реквизиты

Автор
Журнал
Статья
Год аттестации
Число статей
Страна
Ученая степень

Отношения

W (Автор, Год аттестации, Число статей)
 T (Автор, Ученая степень)
 P (Автор, Журнал, Статья)
 R (Журнал, Страна)



Названия журналов не повторяются.

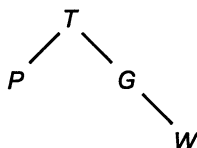
Задание 2.92. Реализуйте иерархическую структуру. Разрешается добавлять или исключать имена реквизитов в отношениях. Разработайте пример записи иерархической базы данных.

Реквизиты

Учреждение
Отдел
Тема
Код оборудования
Фιο сотрудника
Продолжительность работы

Отношения

T (Учреждение, Отдел)
 G (Отдел, Тема)
 P (Фιο, Отдел)
 W (Тема, Код оборудования, Продолжительность)



Каждая тема выполняется в единственном отделе.

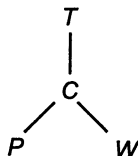
Задание 2.93. Реализуйте иерархическую структуру. Разрешается добавлять или исключать имена реквизитов в отношениях. Разработайте пример записи иерархической базы данных.

Реквизиты

Город
Страна
Команда
Фιο игрока
Год рождения
Дата
Команда-соперник
Счет

Отношения

T (Город, Страна)
 C (Команда, Город)
 P (Фιο, Команда, Год рождения)
 W (Дата, Команда-соперник, Счет)



Названия команд не повторяются.

Моделирование предметных областей

Методические указания

1. Модель «сущность – связь».

Модель «сущность – связь» – это графическая модель, отражающая взаимосвязи между объектами («сущностями») в предметной области. Предметной областью называются элементы материальной системы, информация о которых хранится и обрабатывается в ЭИС. Для описания предметной области необходимы четыре понятия – объект («сущность»), свойство объекта, взаимодействие объектов («связь»), свойство взаимодействия.

В экономических приложениях понятие «объект» сужается до понятия «физический объект», под которым понимается любой предмет, занимающий место в пространстве.

Классификация экономических объектов начинается с выделения средств производства, предметов труда и исполнителей.

Свойством объекта называется величина, которая характеризует объект в каждый момент времени. Определение объекта сводится к выделению его свойств. Два объекта являются различными, если они отличаются по значению хотя бы одного свойства.

Взаимодействием объектов называется факт участия нескольких объектов в каком-либо процессе, который протекает и во времени и в пространстве.

Свойством взаимодействия называется такое свойство, которое характеризует совместное поведение объектов, но не относится ни к одному объекту в отдельности.

На рис. 2.11 показан пример модели «сущность – связь» для описания процесса производства деталей. В овалах представлены объекты и ниже – их свойства. В ромбе указано название взаимодействия и ниже – свойства взаимодействия. Объекты и соответствующее взаимодействие связаны стрелками.

Среди свойств объекта необходимо выделять идентифицирующее свойство, по значениям которого различаются любые два объекта данного класса (двое рабочих, две детали).

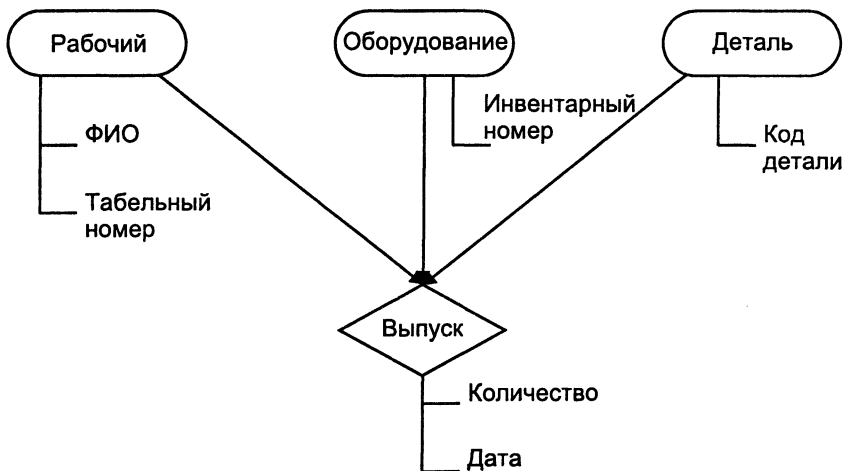


Рис. 2.11. Модель «сущность – связь» для описания процесса производства деталей

В базе данных экономической информационной системы создается по одному отношению для каждого объекта и по одному отношению для каждого взаимодействия. В случае взаимодействия приводятся идентифицирующие свойства соответствующих объектов и свойства взаимодействия.

Согласно рис. 2.11 будут получены отношения:

$T1(\text{Табельный номер, ФИО});$

$T2(\text{Табельный номер, Инвентарный номер, Код детали, Дата, Количество}).$

2. Логический вывод.

Логический вывод реализуется в рамках продукционной модели знаний. Продукционная модель знаний предполагает наличие рабочей памяти, в которой хранятся знания в виде фактов и набора правил. Факт представляет собой любое утверждение. Правила логического вывода имеют вид условных предложений:

Если <условие 1> и <условие 2> и ...

То <результатирующее условие>.

Существуют два механизма логического вывода.

1. Прямой вывод фактов.

В прямом выводе имеющиеся правила применяются к фактам, записанным в рабочей памяти. Если условия, приведенные в левой части некоторого правила, соблюдаются, то результирующее условие в виде факта добавляется в рабочую память, а само правило в дальнейшем не используется. Список правил каждый раз рассматривается сначала.

2. Обратный вывод.

При обратном выводе фиксируется факт, представляющий цель вывода, и отыскиваются правила и факты, подтверждающие истинность цели. Эти факты становятся новыми целями, и процесс повторяется. Обратный вывод заканчивается, когда доказаны все промежуточные цели.

Рассмотрим пример.

П р а в и л а:

1. Если метод доступа – индексный, то СУБД – Access.
2. Если метод доступа – последовательный, то СУБД – Access.
3. Если доля выборки $< 0,1$, то метод доступа – индексный
4. Если СУБД – Access и ЭВМ – Celeron, то программист –

Иванов.

Ф а к т ы:

1. Доля выборки = 0,09.
2. ЭВМ – Celeron.

Последовательность применяемых правил при прямом выводе фактов показана ниже.

Применяемое правило:

3

1

4

Остановка

Новые факты:

метод доступа – индексный

СУБД – Access

программист – Иванов

Последовательность применяемых правил и целей при обратном выводе для цели «программист – Иванов» выглядит следующим образом:

Новая цель:

Программист – Иванов

ЭВМ – Celeron

СУБД – Access

метод доступа – индексный

доля выборки $< 0,1$

Доказательство:

Правило 4

Факт 2

Правило 1

Правило 3

Факт 1

Задание 2.94. Создайте графические представления сущностей и связей для исходных множеств реквизитов, приведенных в заданиях 1.27 – 1.52, 2.33 – 2.49, 2.74 – 2.88. При необходимости разрешается добавлять или удалять отдельные реквизиты.

Задание 2.95. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 1 и 2.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для приведенных ниже фактов и правил, если необходимо доказать цель с номером 5.

П р а в и л а :

1. если ЦЕЛЬ – ОТДЫХ и ДОРОГА ПЛОХАЯ то ИСПОЛЬЗОВАТЬ ДЖИП.

2. если МЕСТО_ОТДЫХА – ГОРЫ то ДОРОГА ПЛОХАЯ.

3. если МЕСТО_ОТДЫХА – ПЛЯЖ то ДОРОГА ПЛОХАЯ.

Ф а к т ы:

1. ЦЕЛЬ – ОТДЫХ.

2. МЕСТО_ОТДЫХА – ГОРЫ.

3. ДОРОГА ПЛОХАЯ.

4. МЕСТО_ОТДЫХА – ПЛЯЖ .

5. ИСПОЛЬЗОВАТЬ ДЖИП.

Задание 2.96. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 1 и 5.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для приведенных ниже фактов и правил, если необходимо доказать цель с номером 2.

П р а в и л а :

1. если КУПИТЬ СКАНЕР и КУПИТЬ ДИСПЛЕЙ то КУПИТЬ МОДЕМ.

2. если КУПИТЬ СКАНЕР и КУПИТЬ ПРИНТЕР то КУПИТЬ СТРИМЕР.

3. если КУПИТЬ КСЕРОКС то КУПИТЬ ДИСПЛЕЙ.

4. если КУПИТЬ МОДЕМ то КУПИТЬ ПРИНТЕР.

Факты:

1. КУПИТЬ СКАНЕР.
2. КУПИТЬ ПРИНТЕР.
3. КУПИТЬ ДИСПЛЕЙ.
4. КУПИТЬ МОДЕМ.
5. КУПИТЬ КСЕРОКС.
6. КУПИТЬ СТРИМЕР.

Задание 2.97. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 1 и 3.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для приведенных ниже фактов и правил, если необходимо доказать цель с номером 2.

Правила:

1. если ФИРМА «D» ПРИБЫЛЬНА то ФИРМА «B» ПРИБЫЛЬНА.
2. если ФИРМА «A» ПРИБЫЛЬНА то ФИРМА «F» ПРИБЫЛЬНА.
3. если ФИРМА «A» ПРИБЫЛЬНА и ФИРМА «C» ПРИБЫЛЬНА то ФИРМА «D» ПРИБЫЛЬНА.
4. если ФИРМА «E» ПРИБЫЛЬНА то ФИРМА «C» ПРИБЫЛЬНА.

Факты:

1. ФИРМА «A» ПРИБЫЛЬНА.
2. ФИРМА «B» ПРИБЫЛЬНА.
3. ФИРМА «C» ПРИБЫЛЬНА.
4. ФИРМА «D» ПРИБЫЛЬНА.
5. ФИРМА «E» ПРИБЫЛЬНА.
6. ФИРМА «F» ПРИБЫЛЬНА.

Задание 2.98. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 1 и 4.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для при-

веденных ниже фактов и правил, если необходимо доказать цель с номером 6.

П р а в и л а :

1. если АКЦИИ «А» ЛИКВИДНЫ и АКЦИИ «В» ЛИКВИДНЫ то АКЦИИ «С» ЛИКВИДНЫ.
2. если АКЦИИ «С» ЛИКВИДНЫ то АКЦИИ «F» ЛИКВИДНЫ.
3. если АКЦИИ «D» ЛИКВИДНЫ то АКЦИИ «В» ЛИКВИДНЫ.
4. если АКЦИИ «А» ЛИКВИДНЫ то АКЦИИ «Е» ЛИКВИДНЫ.

Ф а к т ы :

1. АКЦИИ «А» ЛИКВИДНЫ.
2. АКЦИИ «В» ЛИКВИДНЫ.
3. АКЦИИ «С» ЛИКВИДНЫ.
4. АКЦИИ «D» ЛИКВИДНЫ.
5. АКЦИИ «Е» ЛИКВИДНЫ.
6. АКЦИИ «F» ЛИКВИДНЫ.

Задание 2.99. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 3 и 6.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для приведенных ниже фактов и правил, если необходимо доказать цель с номером 4.

П р а в и л а :

1. если ЗАПАСЫ ДОСТАТОЧНЫ то ЛИКВИДНОСТЬ ХОРОШАЯ.
2. если ЛИКВИДНОСТЬ ХОРОШАЯ и ПЛАТЕЖЕСПОСОБНОСТЬ ХОРОШАЯ то ОБЯЗАТЕЛЬСТВА МИНИМАЛЬНЫ.
3. если ОСТАТКИ СРЕДСТВ ДОСТАТОЧНЫ то УСТОЙЧИВОСТЬ ХОРОШАЯ.
4. если ОБЯЗАТЕЛЬСТВА МИНИМАЛЬНЫ то УСТОЙЧИВОСТЬ ХОРОШАЯ.

Ф а к т ы :

1. ЛИКВИДНОСТЬ ХОРОШАЯ.
2. ОСТАТКИ СРЕДСТВ ДОСТАТОЧНЫ.

3. ПЛАТЕЖЕСПОСОБНОСТЬ ХОРОШАЯ.
4. УСТОЙЧИВОСТЬ ХОРОШАЯ.
5. ОБЯЗАТЕЛЬСТВА МИНИМАЛЬНЫ.
6. ЗАПАСЫ ДОСТАТОЧНЫ.

Задание 2.100. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 5 и 6.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для приведенных ниже фактов и правил, если необходимо доказать цель с номером 3.

П р а в и л а :

1. если АКЦИИ АО «ТЕЗА» ЛИКВИДНЫ то ИНВЕСТИРОВАТЬ АО «РИТМ».

2. если АКЦИИ АО «РИТМ» ЛИКВИДНЫ то ИНВЕСТИРОВАТЬ АО «РИТМ».

3. если КУРС ДОЛЛАРА ПАДАЕТ то АКЦИИ АО «ТЕЗА» ЛИКВИДНЫ.

4. если ИНВЕСТИРОВАТЬ АО «РИТМ» и ДИРЕКТОР АО «РИТМ» – ПЕТРОВ то ИНВЕСТИРОВАТЬ АО «ТЕЗА».

Ф а к т ы :

1. АКЦИИ АО «ТЕЗА» ЛИКВИДНЫ.

2. АКЦИИ АО «РИТМ» ЛИКВИДНЫ.

3. ИНВЕСТИРОВАТЬ АО «ТЕЗА».

4. ИНВЕСТИРОВАТЬ АО «РИТМ».

5. КУРС ДОЛЛАРА ПАДАЕТ.

6. ДИРЕКТОР АО «РИТМ» – ПЕТРОВ.

Задание 2.101. Установите последовательность применяемых правил и новых фактов в процессе работы механизма прямого вывода для приведенных ниже фактов и правил, если первоначально в рабочей памяти хранятся факты с номерами 5 и 1.

Установите последовательность новых целей и их доказательств в процессе работы механизма обратного вывода для приведенных ниже фактов и правил, если необходимо доказать цель с номером 6.

П р а в и л а :

1. если ИНФЛЯЦИЯ СНИЗИЛАСЬ то КРЕДИТ ВОЗВРАЩЕН.

2. если ДИВИДЕНДЫ ВЫПЛАЧЕНЫ то ИНВЕСТИЦИИ ПОЛУЧЕНЫ.

3. если ЦЕНЫ СТАБИЛЬНЫ и КРЕДИТ ВОЗВРАЩЕН то ДИВИДЕНДЫ ВЫПЛАЧЕНЫ.

4. если ЦЕНЫ СТАБИЛЬНЫ и КРЕДИТ ВОЗВРАЩЕН то НАЛОГИ УПЛАЧЕНЫ.

Ф а к т ы :

1. ЦЕНЫ СТАБИЛЬНЫ.

2. НАЛОГИ УПЛАЧЕНЫ.

3. КРЕДИТ ВОЗВРАЩЕН.

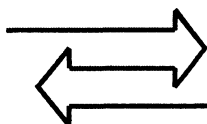
4. ДИВИДЕНДЫ ВЫПЛАЧЕНЫ.

5. ИНФЛЯЦИЯ СНИЗИЛАСЬ.

6. ИНВЕСТИЦИИ ПОЛУЧЕНЫ.

Какие правила, на ваш взгляд, являются экономически необоснованными?

Замените их и повторите решение.



АЛГОРИТМЫ И ОРГАНИЗАЦИЯ ДАННЫХ

3.1.

Последовательная организация данных

Методические указания

При последовательной организации данных записи располагаются в памяти строго одна за другой, без промежутков, в той последовательности, в которой они обрабатываются. Последовательная организация данных соответствует понятию «массив» (файл).

В структуре записей последовательного массива выделяется один или несколько ключевых реквизитов, по значениям которых происходит доступ к остальным значениям реквизитов той или иной записи. Состав ключевых реквизитов необязательно соответствует понятию «первичный ключ».

Будем считать, что последовательный массив состоит из записей фиксированной длины, а среди реквизитов записи выделяется только один ключевой реквизит. Наличие в записях других (неключевых) реквизитов специально не оговаривается (рис. 3.1).

$p(1)$	$p(2)$	$p(3)$...	$p(i)$...	$p(M)$
1	2	3		i		M
Номера записей						

Рис. 3.1. Последовательная организация данных

Ключевые реквизиты в записях обозначаются через $p(i)$, где i – номер записи, общее число записей в массиве обозначается через M .

Записи массива могут быть упорядоченными или неупорядоченными по значениям ключевого реквизита (ключа), имя которого одинаково во всех записях. Ключевой реквизит обычно является реквизитом-признаком. Часто требуется поддерживать упорядоченность записей по нескольким именам ключевых признаков. В этом случае среди признаков устанавливается старшинство. Условие упорядоченности записей в массиве (и вообще для линейной организации данных) выглядит следующим образом:

$$p(i) \leq p(i+1) - \text{упорядоченность по возрастанию};$$
$$p(i) \geq p(i+1) - \text{упорядоченность по убыванию}.$$

Наиболее часто применяемыми алгоритмами обработки данных являются формирование данных, их поиск и корректировка, а также последовательная обработка. Эти алгоритмы могут быть реализованы с использованием достаточно большого количества методов организации данных. Сами методы организации данных будут представлены в их простейшей форме.

Поиском называется процедура выделения из некоторого множества записей определенного подмножества, записи которого удовлетворяют некоторому заранее поставленному условию. Условие поиска часто называют запросом на поиск. Простейшим условием поиска является поиск по совпадению, т.е. равенство значения ключевого атрибута i -й записи $p(i)$ и некоторого заранее заданного значения q . Алгоритмы всех разновидностей поиска можно получить из алгоритмов поиска по совпадению, которые и рассматриваются в дальнейшем.

Базовым методом доступа к массиву является ступенчатый поиск. Этот метод предполагает упорядоченность обрабатываемых записей, причем безразлично, по возрастанию или по убыванию. Для определенности будем считать, что массив отсортирован по возрастанию значений ключевого атрибута $p(i)$.

Простейшим вариантом ступенчатого поиска (его можно назвать одноступенчатым) является последовательный поиск. Искомое значение q сравнивается с ключом первой записи, если значения не совпадают, с ключом второй записи и т.д. до тех пор, пока значение q не станет больше ключа очередной записи.

Число сравнений может быть выражено как $C(M) = \sum i \cdot r(i)$, где суммирование по i ведется в пределах от 1 до M .

Поиск с одинаковой вероятностью $r(i)=1/M$ может окончиться на любой записи, поэтому

$$C(M) = 1/M * \sum i = (M + 1)/2 \text{ или } C(M) \sim M.$$

Знак \sim означает, что $C(M) \approx k * M$, где коэффициент пропорциональности k заранее неизвестен.

Ступенчатый поиск имеет важный частный вариант – бинарный поиск.

Для бинарного поиска вводятся левая граница интервала поиска A и правая граница B . Первоначально интервал охватывает весь массив, т.е. $A=0$, $B=M+1$. Вычисляется середина интервала i по формуле $i=(A+B)/2$ с округлением в меньшую сторону. Ключ i -й записи $p(i)$ сравнивается с искомым значением q . Если $p(i)=q$, то поиск заканчивается. В случае $p(i)>q$ записи с номерами $i+1$, $i+2$, ..., M заведомо не содержат ключа q и надо сократить интервал поиска, приняв $B=i$. Аналогично при $p(i)<q$ надо взять $A=i$. Затем середина интервала вычисляется заново, и все действия повторяются. Если будет достигнут нулевой интервал, то требуемой записи в массиве нет.

Достаточно легко оценить максимальное число сравнений $Cm(M)$ при поиске данных бинарным методом. Сокращение интервала поиска на каждом шаге в худшем случае приведет к интервалу нулевой длины, что соответствует отсутствию в массиве искомого значения ключевого атрибута. После первого сравнения интервал поиска составит $M/2$ записей; после второго – $M/4$ и т.д. Когда интервал поиска впервые станет меньше 1, применяемая схема округления результата деления даст нулевой интервал и поиск закончится. Это соответствует неравенству

$$M/(2^{Cm(M)}) \leq 1$$

(знак \wedge обозначает возведение в степень), откуда

$$Cm(M) \sim \log M.$$

Данные обычно возникают в неупорядоченной форме. Перед обработкой, как правило, целесообразно упорядочить их по значениям ключевых реквизитов, что составляет одну из основных работ по формированию (подготовке) данных. Процедуру упорядочения файла часто называют сортировкой.

Рассмотрим некоторые методы сортировки.

1. Сортировка включением. Рассмотрим исходный массив 45, 55, 12, 42, 94, 18, 6, 67. При сортировке включением массив делится на две части: левая уже отсортирована, а правая еще нет. Сначала левая часть состоит из одного элемента, а правая из всех остальных. Необходимо взять первый элемент из правой части (55) и поместить его в левую часть таким образом, чтобы левая часть стала упорядоченной. После этого левая отсортированная часть увеличится на один элемент, а правая на один элемент уменьшится. В нашем примере $45 < 55$, поэтому не надо производить перестановок элементов. Далее вышеописанную процедуру надо повторить с новыми правой и левой частями массива. Видно, что для того чтобы поместить 12 на нужное место, необходимо произвести перестановки элементов. Делается это следующим образом: сравнивается первый элемент правой части и последний элемент левой части (12 и 55). Имеем $12 < 55$, поэтому значение 12 запоминается во временную переменную, а на место 12 помещается 55. Затем 12 сравнивается с 45, 45 помещается на место 55. Эта процедура повторяется до тех пор, пока не будет достигнут левый край массива, либо очередной элемент в левой части массива не окажется меньше 12. После этого 12 помещается на освободившееся место (в данном случае на место 45). Теперь массив разбит следующим образом:

12, 45, 55, | 42, 94, 18, 6, 67.

Вертикальная черта разделяет левую и правую части массива. Следует продолжать действовать подобным образом до тех пор, пока в правой части еще есть элементы.

2. Метод обменной сортировки. Мы будем просматривать весь массив «снизу вверх» и менять стоящие рядом элементы в том случае, если «нижний» элемент меньше, чем «верхний». Таким образом, мы вытолкнем наверх самый «легкий» элемент всего массива. Теперь повторим всю операцию для оставшихся неотсортированными $M - 1$ элементов (т.е. для тех, которые лежат «ниже» первого).

3. Сортировка выбором. На этот раз при просмотре массива мы будем искать наименьший элемент, сравнивая его с первым. Если такой элемент найден, поменяем его местами с первым. Затем повторим эту операцию, но начнем не с первого элемента, а

со второго и будем продолжать подобным образом, пока не сортируем весь массив.

4. Метод Шелла. Метод состоит в том, что сортируемый массив разделяется на ряд групп, каждая из которых сортируется методом включения. В процессе сортировки размеры групп возрастают до тех пор, пока все записи не войдут в одну упорядоченную группу.

Группой массива называют последовательность элементов, номера которых образуют арифметическую прогрессию с шагом d . В начале процесса выбирается шаг $d_1 < M$. Затем определяются группы, элементы которых имеют номера $i, i + d_1, i + 2*d_1, \dots$ при $i = 1, 2, \dots, d_1$. Производится сортировка групп. Выбирается шаг $d_2 < d_1$ и сортируются соответствующие группы. Описанная процедура выполняется до тех пор, пока очередной шаг не станет равен 1. В этот момент происходит упорядочение всего массива методом включения. Выбор шагов производится по следующим формулам:

$$d_1 = \text{int}(M/2); d_i = \text{int}(d_{i-1}/2),$$

где int обозначает округление до целого в меньшую сторону.

Рассмотрим последовательность из 9 элементов: 45, 55, 12, 42, 94, 18, 6, 67, 14. Вначале $d_1 = 4$. Сортируем 1, 5 и 9 элемент, затем 2 и 6, 3 и 7, 4 и 8. В результате получаем последовательность: 14, 18, 6, 42, 45, 55, 12, 67, 94. Следующий шаг $d_2 = 2$. Сортируем 1, 3, 5, 7, 9 элемент, затем 2, 4, 6 и 8. Новое состояние массива имеет вид: 6, 18, 12, 42, 14, 55, 45, 67, 94. После сортировки с шагом $d_3 = 1$ получаем окончательный результат: 6, 12, 14, 18, 42, 45, 55, 67, 94.

Ниже рассмотрены некоторые общие характеристики поисковых алгоритмов, а также некоторые свойства технологической реализации операции поиска, позволяющие оптимизировать характеристики этой операции. Рассмотрим понятие «нагрузка поиска», которое позволяет сравнивать поисковые технологии с точки зрения их эффективности.

Допустим, что в файле размером M записей производится поиск данных по некоторому запросу q , причем известно, что количество релевантных записей равняется m_r .

Рассмотрим для реализации некоторого запроса численную характеристику, соответствующую доле релевантных записей в

файле. Эту величину принято называть *коэффициентом активности* и определять как

$$\Lambda = \frac{m_r}{M},$$

где m_r – количество релевантных записей;
 M – общее количество записей в файле.

При этом количество просматриваемых в процессе поиска записей может составить m_r , $m_r \leq M$.

Введем понятие «нагрузка поиска». Под нагрузкой поиска будем понимать величину Ψ , определяемую как $\Psi = m_l / m_r \geq 1$, т.е. количество записей, которое надо в среднем просмотреть в данном алгоритме поиска на одну найденную релевантную запись.

Пусть $\Lambda = m_r / M \leq 1$ – это коэффициент активности поискового файла относительно данного запроса. Тогда величина

$$\Psi \cdot \Lambda = (m_l / m_r) \cdot (m_r / M) = m_l / M$$

должна находиться в пределах $1/M \leq \Psi \cdot \Lambda \leq 1$. Нижний предел $\Psi \times \Lambda = 1/M$ соответствует тому идеальному случаю, когда на каждую релевантную запись просматривается только одна нагрузочная запись. Нарушение верхнего предела ($\Psi \cdot \Lambda > 1$) означает, что $m_l > M$, т.е. количество нагрузочных записей превышает размер файла. Разумеется, алгоритм поиска с подобным соотношением не имеет смысла использовать, поскольку он менее эффективен, чем метод последовательного поиска.

Рассмотрим один из традиционных, хорошо разработанных и широко применяемых методов снижения нагрузки поиска – метод *индексирования* файла в жестком формате. Приведем, наряду с понятием «индексирование», некоторые другие связанные с ним понятия.

Последовательный доступ – это метод реализации операции выборки, при которой последовательно перебираются все записи файла и на каждой записи проверяется поисковое условие.

Прямой доступ – это такой способ реализации выборки, при котором просмотр производится не во всей базе данных, а в некоторой ее части. Идеальный случай прямого доступа – когда просматриваются только релевантные записи.

Индексирование – это операция, предназначенная для форматированных баз данных, при которой строится вспомогательный файл (файл индексов для организации прямого доступа к основному файлу данных).

В простейшем случае форматированная база данных индексируется только по одному реквизиту. При этом основной файл данных обычно упорядочен по реквизиту, называемому ключом индексирования.

Сокращение времени поиска достигается благодаря тому, что индексный файл короче основного, кроме того, над индексным файлом можно построить индекс второго уровня и получить выигрыш, соответствующий рассматриваемой схеме.

Схема индексирования, в которой индексируются участки файла, называется *индексно-последовательной*.

Схема индексирования, содержащая в индексе столько же записей, сколько в основном файле, называется *индексно-прямой*.

Индексно-последовательную схему можно применять только в том случае, если индексируемый файл упорядочен по значениям ключа индексирования, а если он не упорядочен, то допустима только индексно-прямая схема.

Для организации доступа к записям основного файла используются индексные файлы. Суть индексных файлов состоит в том, что по ключу индексирования устанавливаются адресные указатели, и поэтому удастся организовать быстрый поиск.

Если исходный файл упорядочен по значениям ключа индексирования, то можно индексировать по физическим адресам записей. Если же исходный файл не упорядочен, то в индексе необходимо помещать указатель на каждую запись, а сам индексный файл можно упорядочить по значениям ключа индексирования.

Пусть имеется файл F с реквизитами A и B , приведенный ниже. Записи файла F упорядочены по значениям реквизита A .

A	B	Физический адрес
010	80	5100
020	10	5110
030	15	5120
040	89	5130
050	36	5140
060	75	5150
070	25	5160

A	B	Физический адрес
080	29	5170
090	45	5180
100	59	5190
110	65	5200
120	98	5210
130	19	5220
140	72	5230
150	34	5240
160	55	5250

Индексный файл $I_0(A)$ для поиска по реквизиту A имеет следующий вид.

A	Физический адрес
040	5130
080	5170
120	5210
160	5250

Допустим, задано следующее поисковое условие: $A = 130$.

При просмотре файла $I_0(A)$ будет установлено, что искомая запись находится между записью 120 и 160, следовательно, между этими физическими адресами данных записей необходимо искать адрес требуемой записи в основном массиве. В среднем, при условии, что поиск осуществляется последовательно, и обращения к каждой из находящихся в файле записей равновероятны, придется просматривать 4 записи. А если бы выполнялся последовательный поиск заданной записи в исходном файле F , то в среднем просматривалось бы 8 записей.

Описание процедуры слияния двух упорядоченных файлов

Рассмотрим процедуру MERGE слияния двух файлов A и B , упорядоченных по возрастанию ключей. На иллюстрациях указаны только ключи записей файлов в виде двузначных десятичных цифровых кодов. Пусть файлы A и B имеют следующий вид (табл. 3.1).

Значения ключевых реквизитов для исходных файлов перед слиянием

Номер записи	1	2	3	4	5	6	7	8
<i>A</i>	11	28	37	51	63	69	86	93
<i>B</i>	15	21	27	31	41	59	75	77

Результирующий файл *C*, также упорядоченный по возрастанию значений ключа записи, будет иметь вид (табл. 3.2).

Таблица 3.2

Значения ключевых реквизитов для результирующего файла после слияния

Номер записи	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<i>C</i>	11	15	21	27	28	31	37	41	51	59	63	69	75	77	86	93

Процедура MERGE получения файла *C* слиянием файлов *A* и *B* состоит из следующих шагов:

Шаг 1. Открываются все три файла, причем указатели записей в них (соответственно *pa*, *pb* и *pc*) устанавливаются на 1. Таким образом, во всех трех файлах текущей становится первая запись. Впредь будем обозначать *ra(pa)*, *rb(pb)*, *rc(pc)* записи соответствующих файлов, определяемые их указателями: *pa*, *pb*, *pc*. Например, *rb(5) = 41*, *rc(11) = 63*.

Шаг 2. Сравниваются ключи текущих записей файлов *A* и *B*, причем запись с меньшим ключом переносится в файл *C*. При этом указатели файла *C* и того файла, из которого взята текущая запись, увеличиваются на 1.

Шаг 3. Шаг 2 повторяется до тех пор, пока не будет достигнут конец файла *A* или *B*. Если достигнут конец одного из файлов *A* или *B*, то остаток (соответственно) файла *B* или *A* переписывается без дальнейших сравнений в файл *C* (с соответствующими изменениями указателей).

Выполнение процедуры можно просмотреть на предложенных примерах, приведенных в табл. 3.1 и 3.2. При сравнении ключей первых записей файлов *A* и *B* в файл *C* попадает сначала запись 11 файла *A*, затем три записи 15, 21, 27 файла *B*. Далее попеременно *C* пополняется записями то из *A*, то из *B*, пока файл *B* не закончится записью 77, после чего записи 86 и 93 файла *A* переносятся в файл *C* уже без сравнений.

Инвертированный массив

Для уменьшения времени доступа к записям основного массива формируется вспомогательный инвертированный массив. Основной массив в общем случае предполагается неформатированным. Записью в инвертированном массиве является так называемая группа. Группа состоит из ключевого признака записи основного массива и упорядоченных начальных адресов записей, содержащих данный признак. Группы упорядочиваются по значениям ключевого признака.

Пример основного массива и соответствующего инвертированного массива имеет следующий вид.

Основной массив

Адрес	Запись
0100	<i>B D</i>
0140	<i>A B C</i>
0180	<i>D B A</i>

Инвертированный массив

<i>A</i>
0140
0180
<i>B</i>
0100
0140
0180
<i>C</i>
0140
<i>D</i>
0100
0180

В инвертированном массиве имеется четыре группы.

Задания

Задание 3.1. Напишите программу последовательного поиска в последовательном неотсортированном массиве реквизитов единственного значения q . Используйте любой доступный вам язык программирования.

Задание 3.2. Напишите программу последовательного поиска в последовательном отсортированном массиве реквизитов единственного значения q . Используйте любой доступный вам язык программирования.

Задание 3.3. Напишите программу последовательного поиска в последовательном неотсортированном массиве реквизитов всех значений q . Используйте любой доступный вам язык программирования.

Задание 3.4. Напишите программу последовательного поиска в последовательном отсортированном массиве реквизитов всех значений q . Используйте любой доступный вам язык программирования.

Задание 3.5. Напишите программу бинарного поиска в последовательном отсортированном массиве реквизитов единственного значения q . Используйте любой доступный вам язык программирования.

Задание 3.6. Напишите программу бинарного поиска в последовательном отсортированном массиве реквизитов всех значений q . Используйте любой доступный вам язык программирования.

Задание 3.7. Определите для последовательного поиска:

- максимальное и среднее число сравнений при поиске в неупорядоченном массиве из M элементов единственного значения q ;
- среднее число сравнений при поиске в упорядоченном массиве из M элементов единственного значения q , когда искомого значения в массиве нет.

Задание 3.8. Определим оболочку поискового запроса как множество всех реквизитов, участвующих в формулировке запроса. Сколько различных оболочек может быть у файла из N реквизитов?

Задание 3.9. Реализует ли приведенная ниже программа алгоритм сортировки простым выбором?

```

PROGRAM SimpleSelect;
  const N = 400;
  var J,I,K : Integer;
      Max,Ind: Integer;
      A      : Array [0..N] of Integer;
BEGIN
  For I:=0 to N do
    begin A[I] := Random (N); Write (A[I]:4) end;
  For J:=N downto 1 do
    begin
      Max := A[J]; Ind := J;
      For I:=J downto 0 do
        If A[I]>Max
          then begin Max := A[I]; Ind := I end;
      If Ind<>J
        then begin
          K := A[Ind]; A[Ind] := A[J];
          A[J] := K
        end
      end;
    For I:=0 to N do Write (A[I]:4)
  END.

```

Задание 3.10. Реализуйте сортировку для приведенного ниже файла по реквизиту «Фамилия» средствами табличного процессора Excel. «Год» обозначает год поступления на работу. Экспортируйте отсортированные данные в таблицу Access.

Фамилия	Год
Aghion	1994
Полковский	1991
Armstrong	1974
Arrow	1981
Borkin	1978
Архангельская	1978
Borkin	1970
Howitt	1992
Sundgren	1992
Sundgren	1978
Wagner	1969
Wagner	1964
Беляев	1977
Буренкова	1980
Келехсаев	1987

Задание 3.11. Напишите программу слияния двух упорядоченных файлов с неодинаковым количеством элементов. Используйте любой доступный вам язык программирования.

Задание 3.12. Напишите программу индексирования основного файла по одному реквизиту. Используйте любой доступный вам язык программирования.

Задание 3.13. Индексируйте приведенный ниже файл по реквизиту *В*. Необходимые дополнительные параметры выберите самостоятельно.

<i>A</i>	<i>B</i>	Физический адрес
020	10	5110
030	15	5120
130	19	5220
070	25	5160
080	29	5170
150	34	5240
050	36	5140
090	45	5180
160	55	5250
100	59	5190
110	65	5200
140	72	5230
060	75	5150
010	80	5100
040	89	5130
120	98	5210

Задание 3.14. Реализуйте описанные выше операции индексирования средствами СУБД Access не менее чем в трех вариантах, изменяя размер файла, размер участка индексирования и метод индексирования.

Задание 3.15. Рассчитайте среднее значение величины $\Psi \cdot \Lambda$ для последовательного поиска, если количество релевантных записей точно известно заранее.

Задание 3.16. Создайте средствами СУБД Access индексную структуру для поиска терминов по предметным областям, и предметных областей – по терминам.

Задание 3.17. Рассмотрите файл из двух реквизитов A и B с первой записью (11,8) и последующими значениями A и B , получаемыми по формулам

$$\begin{aligned} A(i+1) &= (7 \cdot A(i) + 19) \bmod 13; \\ B(i+1) &= (5 \cdot B(i) + 11) \bmod 17, \end{aligned}$$

состоящий из 25 записей. Создайте индексные файлы по реквизитам A и B и двум реквизитам совместно. Необходимые дополнительные параметры выберите самостоятельно.

Задание 3.18. Дан основной массив. Каждая запись имеет по два ключевых признака из множества признаков $A = \{a, b, c, d, e, f, g, h\}$. Признак имеет длину 4 байта, адрес записи – 8 байт. Определить объем памяти под инвертированный массив.

Задание 3.19. Представьте схему инвертированного массива для следующего массива записей.

Адрес	Ключевые признаки
100	$E B D$
140	$C E A$
200	$D E$
300	$C E$

3.2.

Цепная организация данных

Методические указания

Списком называется множество записей, занимающих произвольные участки памяти, последовательность обработки которых задается с помощью адресов связи. *Адресом связи* некоторой записи называется реквизит, в котором хранится начальный адрес или номер записи, обрабатываемой после этой записи. Последовательность обработки записей в списке определяется возрастанием значений ключа в записях.

В списке выделяется собственная информация (записи с содержательными сведениями) и ассоциативная информация, т.е. все адреса связи.

Описание записей списка на языке программирования (например, Паскаль) может быть проведено двумя способами.

1. Определение адресов связи как начальных адресов записей:

```
type ref=^node;
node=record
  key: integer; {ключевой реквизит записи}
  other: string[30]; {остальные реквизиты}
  next: ref {адрес связи}
end;
```

2. Определение адресов связи как номеров записей:

```
const M=12;
type
node=record
  key: integer; {ключевой реквизит записи}
  other: string[30]; {остальные реквизиты}
  next: integer {адрес связи}
end;
var t:array[1..M] of node;
```

Второй вариант является более практичным, особенно если требуется хранить список на внешнем запоминающем устройстве.

При списковой организации данных необходим специальный реквизит, называемый указателем списка, который содержит начальный адрес или номер первой в порядке обработки записи списка. Кроме того, адрес связи последней записи списка должен содержать специальное значение, называемое концом списка и отмечающее, что последующих записей у данной записи нет. Обычно конец списка обозначается нулем.

Графически адрес связи изображается в виде прямоугольника со стрелкой, стрелка указывает на запись, адрес хранения которой содержится в адресе связи.

При формировании упорядоченного списка записей возможны два варианта:

- вновь поступающие записи вставлять так, чтобы не нарушать упорядоченность по ключу;
- создать сначала неупорядоченный список, а затем отсортировать его.

Учитывая, что для сортировки можно использовать метод слияния, второй вариант следует признать более целесообразным.

В итоге время формирования упорядоченного списка пропорционально $T \sim M \cdot \log M$.

Для поиска данных в упорядоченном списке можно применять те же методы, что и в последовательном массиве, однако эффективность этих методов иная, поскольку адреса связи создают возможность быстрого доступа только к следующей записи списка.

Для поиска данных в однонаправленном списке используется единственный метод – последовательный поиск. Ключевой реквизит первой записи (ее адрес извлекается из указателя списка) сравнивается с искомым значением q , затем такое же сравнение выполняется для ключа второй записи, которая извлекается по адресу связи первой записи, и т.д. Время поиска, естественно, пропорционально $T \sim M$.

Цепным каталогом называется сплошной участок памяти (или несколько таких участков), в котором одновременно размещаются список обрабатываемых записей и список свободных позиций памяти (УСП). Адрес связи, отмечающий первую обрабатываемую запись, называется указателем списка (УС). Адрес связи, отмечающий первую свободную позицию памяти, называется указателем свободной памяти. Адрес связи последней записи (или последней позиции свободной памяти) в списке называется концом списка, и здесь отмечается нулевым значением.

Рассмотрим пример цепного каталога, в котором адреса связи представлены номерами соответствующих записей. Описание каталога на языке Паскаль имеет вид:

```
const M=9;
type
node=record
    key: integer; {ключевой реквизит записи}
    next: integer {адрес связи}
end;
var t:array[1..M] of node;
```

Первоначальное состояние каталога показано на рис. 3.2 а.

Включение и исключение записей в цепном каталоге предполагает поиск местоположения включаемой (исключаемой) записи и замену значений адресов связи для установления новой последовательности записей основного списка и списка свободной памяти.

Приведем алгоритм вставки записи с ключом F в цепной каталог.

1. Найти в каталоге запись с ключом непосредственно меньше, чем F (предшествующая запись).
2. Поместить запись с ключом F в первую позицию свободной памяти.
3. Заменить УСП на адрес связи новой записи, этот адрес – на адрес связи предшествующей записи, а последний – на первоначальное значение УСП.

На рис. 3.2 *а* вставляется запись с ключом $F=61$.

Алгоритм удаления записи с ключом W из каталога.

1. Найти в каталоге запись с ключом непосредственно меньше, чем W (предшествующая запись).
2. Заменить УСП на адрес связи предшествующей записи, этот адрес – на адрес связи исключаемой записи, а последний – на первоначальное значение УСП.

На рис. 3.2 *б* удаляется запись с ключом $W=30$.

	KEY	NEXT	US = 6 USP = 3
1	50	5	
2	20	8	
3		7	
4	40	1	
5	60	9	
6	10	2	
7		0	
8	30	4	
9	70	0	

До вставки записи

	KEY	NEXT	US = 6 USP = 7
1	50	5	
2	20	8	
3	61	9	
4	40	1	
5	60	3	
6	10	2	
7		0	
8	30	4	
9	70	0	

После вставки записи

а

	KEY	NEXT	US = 6 USP = 3
1	50	5	
2	20	8	
3		7	
4	40	1	
5	60	9	
6	10	2	
7		0	
8	30	4	
9	70	0	

До удаления записи

	KEY	NEXT	US = 6 USP = 8
1	50	5	
2	20	4	
3		7	
4	40	1	
5	60	9	
6	10	2	
7		0	
8	30	3	
9	70	0	

После удаления записи

б

Рис. 3.2. Корректировка данных в цепном каталоге

Оценка времени корректировки складывается из времени реализации поиска и времени на замену значений адресов связи. В последнем случае число пересылок адресов связи всегда одинаково и не зависит от числа записей в цепном каталоге, поэтому затраты времени на поиск при корректировке являются доминирующими и время корректировки пропорционально $T \sim M$.

Задания

Задание 3.20. Установите адреса связи и указатели для обработки записей по возрастанию значений ключа.

УС 0		
УСП 0		
Адрес записи	Значение ключа	Адрес связи
10	375	
11	215	
12	97	
13	115	
14		
15	35	
16	400	

Задание 3.21. Установите адреса связи и указатели для обработки записей по возрастанию значений ключа.

УС 0		
УСП 0		
Адрес записи	Значение ключа	Адрес связи
10	28	
11	45	
12	64	
13	15	
14	31	
15		
16	98	

Задание 3.22. Установите адреса связи и указатели для обработки записей по возрастанию значений ключа.

УС 0		
УСП 0		
Адрес записи	Значение ключа	Адрес связи
10		
11	16	
12	37	
13	85	
14	19	
15	33	
16	91	

Задание 3.23. Установите адреса связи и указатели для удаления записи с ключом 45 из цепного каталога.

УС 11		
УСП 10		
Адрес записи	Значение ключа	Адрес связи
10		0
11	16	14
12	37	13
13	85	16
14	19	15
15	33	12
16	91	0

Задание 3.24. Установите адреса связи и указатели для обработки записей по возрастанию значений ключа.

УС 0		
УСП 0		
Адрес записи	Значение ключа	Адрес связи
10	15	
11	20	
12	10	
13	18	
14	25	
15	36	
16		

Задание 3.25. Установите адреса связи и указатели для удаления записи с ключом 45 из цепного каталога.

УС 13		
УСП 15		
Адрес записи	Значение ключа	Адрес связи
10	28	14
11	45	12
12	64	16
13	15	10
14	31	11
15		0
16	98	0

Задание 3.26. Установите адреса связи и указатели для вставки записи с ключом 16 в цепной каталог.

УС 12		
УСП 16		
Адрес записи	Значение ключа	Адрес связи
10	15	13
11	20	14
12	10	10
13	18	11
14	25	15
15	36	0
16		0

Задание 3.27. Установите адреса связи и указатели для удаления записи с ключом 35 из цепного каталога.

УС 15		
УСП 14		
Адрес записи	Значение ключа	Адрес связи
10	375	16
11	215	10
12	97	13
13	115	11
14		0
15	35	12
16	400	0

Задание 3.28. Установите адреса связи и указатели для удаления записи с ключом 18 из цепного каталога.

УС 12 УСП 16		
Адрес записи	Значение ключа	Адрес связи
10	15	13
11	20	14
12	10	10
13	18	11
14	25	15
15	36	0
16		0

Задание 3.29. Установите адреса связи и указатели для обработки записей по возрастанию значений ключа.

УС 0 УСП 0		
Адрес записи	Значение ключа	Адрес связи
10	20	
11	14	
12	41	
13		
14	12	
15	30	
16	54	

Задание 3.30. Установите адреса связи и указатели для вставки записи с ключом 32 в цепной каталог.

УС 14 УСП 13		
Адрес записи	Значение ключа	Адрес связи
10	20	15
11	14	10
12	41	16
13		0
14	12	11
15	30	12
16	54	0

Задание 3.31. Установите адреса связи и указатели для удаления записи с ключом 14 из цепного каталога.

УС 14		
УСП 13		
Адрес записи	Значение ключа	Адрес связи
10	20	15
11	14	10
12	41	16
13		0
14	12	11
15	30	12
16	54	0

Задание 3.32. Установите адреса связи и указатели для обработки записей по возрастанию значений ключа.

УС 0		
УСП 0		
Адрес записи	Значение ключа	Адрес связи
10	3	
11	91	
12	13	
13	45	
14	51	
15	25	
16		

Задание 3.33. Установите адреса связи и указатели для вставки записи с ключом 80 в цепной каталог.

УС 10		
УСП 16		
Адрес записи	Значение ключа	Адрес связи
10	3	12
11	91	0
12	13	15
13	45	14
14	51	11
15	25	13
16		0

Задание 3.34. Установите адреса связи и указатели для удаления записи с ключом 13 из цепного каталога.

УС 10		
УСП 16		
Адрес записи	Значение ключа	Адрес связи
10	3	12
11	91	0
12	13	15
13	45	14
14	51	11
15	25	13
16		0

Задание 3.35. Установите адреса связи и указатели для вставки записи с ключом 85 в цепной каталог.

УС 15		
УСП 14		
Адрес записи	Значение ключа	Адрес связи
10	375	16
11	215	10
12	97	13
13	115	11
14		0
15	35	12
16	400	0

Задание 3.36. Установите адреса связи и указатели для вставки записи с ключом 75 в цепной каталог.

УС 13		
УСП 15		
Адрес записи	Значение ключа	Адрес связи
10	28	14
11	45	12
12	64	16
13	15	10
14	31	11
15		0
16	98	0

Задание 3.37. Установите адреса связи и указатели для вставки записи с ключом 38 в цепной каталог.

		УС	11
		УСП	10
Адрес записи	Значение ключа	Адрес связи	
10		0	
11	16	14	
12	37	13	
13	85	16	
14	19	15	
15	33	12	
16	91	0	

3.3.

Древовидная организация данных

Методические указания

Древовидной организацией данных (деревом) называется множество записей, расположенных по уровням следующим образом:

на 1-м уровне расположена только одна запись (корень дерева), к любой записи i -го уровня ведет адрес связи только от одной записи уровня $i-1$. В данном определении понятия «дерево» и «уровень» вводятся одновременно. Если записи получают номера уровней, соответствующие определению, то они получают и древовидную организацию.

Количество уровней в дереве называется *рангом*. Записи дерева, которые адресуются от общей записи $(i-1)$ -го уровня, образуют группу. Максимальное число элементов в группе называется *порядком* дерева.

Рассмотрим деревья порядка 2 (бинарные). Они интересны тем, что составляющие их записи могут быть упорядочены, для чего один из реквизитов записи должен быть объявлен ключевым.

Для того чтобы определить понятие «упорядоченность» бинарных деревьев, требуется ввести ряд новых понятий. В качестве примера рассмотрим бинарное дерево, изображенное на

рис. 3.3 (внутри показаны значения ключевого реквизита). Запись *A* – корень дерева. Записи, у которых заполнены два адреса связи, называются полными, записи с одним заполненным адресом – неполными, записи с двумя незаполненными адресами – концевыми. Каждая запись имеет левую и правую ветви. Правую (левую) ветвь записи образует поддереву, адресованное из этой записи через правый (левый) адрес связи. У записи *C* правая ветвь состоит из записей *F*, *I*, *J*, левая ветвь – пустая.

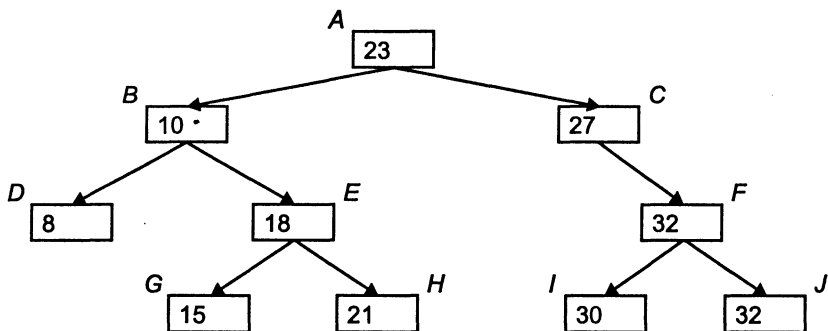


Рис. 3.3. Пример бинарного упорядоченного дерева

В упорядоченном бинарном дереве значение ключевого реквизита каждой записи должно быть больше, чем значение ключа у любой записи на ее левой ветви, и не меньше, чем ключ любой записи на ее правой ветви. Это определение позволяет также различать левые и правые адреса (ветви).

Упорядоченное бинарное дерево формируется из неупорядоченного массива записей по специальному алгоритму. Этот алгоритм создает дерево из первой записи массива, затем – дерево из первых двух записей, из первых трех записей и так далее до исчерпания всех записей массива.

Первая запись массива с ключом p_1 становится корнем дерева. Значение ключа второй записи p_2 сравнивается с p_1 , находящимся в корне дерева. Если $p_2 < p_1$, то вторая запись помещается на левой от корня ветви, в противном случае – на правой ветви. Таким образом, получено упорядоченное дерево из первых двух записей, далее на каждом шаге создается упорядоченное дерево из первых i записей. Выбор места i -й записи массива проводится следующим образом. Ключ p_i сравнивается с корневым значени-

ем, и выполняется переход по левому адресу (если $p1 > pi$), а при $p1 \leq pi$ – по правому адресу. Ключ достигнутой записи также сравнивается с pi , и снова организуется переход по левому или правому адресу и т.д. Когда будет достигнут незаполненный адрес связи, он должен адресовать запись с ключом pi . Указанные действия повторяются до исчерпания всех записей исходного массива.

Упорядоченное бинарное дерево на рис. 3.3 получается из массива с ключевыми реквизитами 23, 10, 18, 27, 15, 32, 8, 30, 32, 21.

В процессе поиска данных по совпадению в упорядоченном бинарном дереве просматривается некоторый путь по дереву, начинающийся всегда в его корне. Искомое значение ключа q сравнивается со значением корня $p1$. Если $p1 > q$, просмотр дерева продолжается по левой ветви корня, если $p1 \leq q$ – по правой. Для произвольной записи дерева с ключом pi результаты сравнения означают:

$pi = q$ – запись, удовлетворяющая условию поиска, найдена, и путь продолжается по правой ветви pi ;

$pi > q$ – производится переход к записи, расположенной на левой ветви pi ;

$pi < q$ – производится переход к записи, расположенной на правой ветви pi .

Поиск заканчивается, когда у какой-либо записи отсутствует адрес связи, необходимый для дальнейшего продолжения поиска. Получаем оценку числа сравнений при поиске в упорядоченном бинарном дереве:

$$C(M) = 1,4 \log M \text{ или } C(M) \sim \log M.$$

При формировании упорядоченного бинарного дерева в среднем производится

$$C(M) = 1,4M \cdot \log M$$

сравнений пар ключевых реквизитов, где M – число записей, для которых строится дерево. Это соответствует затратам на поиск местоположения очередной записи в упорядоченном бинарном дереве из двух, трех и до M записей.

Включение новой записи при корректировке упорядоченного бинарного дерева означает выполнение одного шага алгоритма формирования дерева с включаемой записью на входе.

Сложность исключения зависит от того, какая запись исключается – концевая, неполная или полная. Первые два случая аналогичны корректировке при списковой организации данных. Адрес связи на исключаемую концевую запись заменяется на признак конца строки, адрес связи на исключаемую неполную запись – на ее собственный адрес связи.

При исключении полной записи решается задача о подстановке на ее место другой записи, такой, что ее ключ не нарушает общей упорядоченности бинарного дерева – подобные записи называются соседними. Соседняя слева запись – это запись с ключом, который непосредственно меньше ключа данной записи, а ключ соседней справа записи – равен или непосредственно больше, чем ключ данной записи. Способ нахождения соседней справа записи очень простой. Если исключаемая запись имеет ключ q , то от нее происходит переход по правой ветви дерева и проводится поиск от достигнутой записи значения q . Запись, на которой остановится поиск, будет соседней. Она пересылается на место исключаемой записи, а сама соседняя запись исключается. Это уже простая задача, так как соседняя запись не может быть полной. При поиске соседней слева записи надо выполнить переход по левой ветви от данной записи (с ключом q), а дальнейшие действия – такие же, как и для поиска соседней справа записи.

Задания

Задание 3.38. Можно ли определить дерево как организацию данных, в которой каждый элемент, кроме корня, имеет только один предшествующий элемент и наличие корня обязательно?

Задание 3.39. Дайте определение корня в древовидной организации данных, не используя понятия «уровень».

Задание 3.40. Всегда ли выполняется неравенство между числом записей в дереве M и произведением порядка дерева p на его ранг?

Задание 3.41. Сколько адресов связи содержат однонаправленное и двунаправленное дерево порядка p , состоящее из M записей?

Задание 3.42. Удовлетворяет ли одиночная запись определению дерева?

Задание 3.43. Если произвольная организация данных содержит контур из адресов связи, существует ли для нее понятие «ранг»?

Задание 3.44. Какой вид имеет упорядоченное бинарное дерево, построенное по указанному выше алгоритму для массива, который отсортирован: а) по возрастанию; б) по убыванию?

Задание 3.45. Могут ли различные неупорядоченные массивы из M записей приводить к упорядоченным бинарным деревьям одинаковой конфигурации?

Задание 3.46. Всегда ли поиск заканчивается в висячей вершине?

Задание 3.47. Сколько соседних элементов имеет каждая запись упорядоченного бинарного дерева?

Задание 3.48. Какая причина может препятствовать построению симметричного бинарного дерева?

Задание 3.49. Существует ли соответствие между упорядоченными бинарными деревьями и цепной организацией данных?

Задание 3.50. Как можно использовать упорядоченные бинарные деревья для подсчета частоты встречаемости слов в тексте?

Задание 3.51. Доказать, что если запись B располагается на правой ветви записи A , а запись C – на левой ветви записи B , то ключ записи C имеет промежуточное значение между ключами A и B .

Задание 3.52. Доказать, что:

- если вновь включаемая в дерево запись B адресуется из записи A , то записи A и B являются соседними друг для друга;
- если записи B и A – соседние, то либо запись B расположена на ветви записи A , либо запись A расположена на ветви записи B .

Задание 3.53. Если один из соседних элементов переставить на место другого, в каких случаях упорядоченность бинарного дерева сохранится, а в каких – нет?

Задание 3.54. Постройте упорядоченные бинарные деревья для пяти массивов со следующими значениями ключевых признаков.

- 1) 56, 46, 39, 76, 49, 97, 75, 39, 8, 59, 36, 80, 15, 46, 61;
- 2) 48, 14, 53, 85, 72, 31, 20, 76, 64, 30, 19, 43, 17, 59, 87;

- 3) 69, 30, 70, 85, 35, 96, 25, 18, 47, 56, 42, 34, 70, 52, 93;
 4) 51, 17, 22, 82, 98, 50, 79, 34, 20, 41, 36, 80, 29, 55, 61;
 5) 34, 47, 61, 53, 27, 74, 13, 30, 55, 50, 23, 47, 28, 15, 32.

Подсчитайте число сравнений при поиске в этих пяти деревьях.

В 1-м варианте – ключевой признак 49.

Во 2-м варианте – ключевой признак 85.

В 3-м варианте – ключевой признак 34.

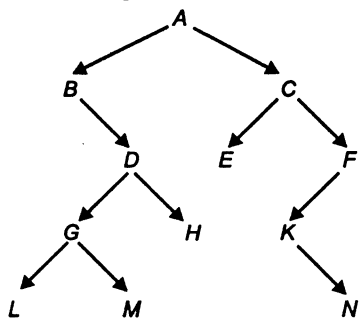
В 4-м варианте – ключевой признак 79.

В 5-м варианте – ключевой признак 53.

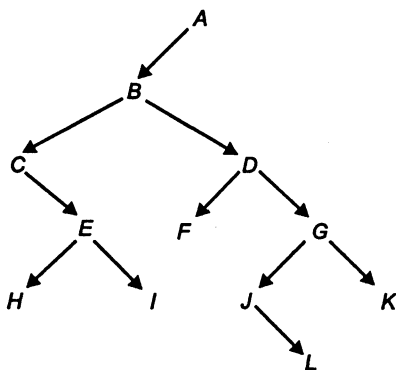
Усредните полученные данные и сделайте выводы.

Задание 3.55. Определите максимальное число полных вершин в симметричном дереве ранга t .

Задание 3.56. Проставьте в вершинах бинарных деревьев (рис. 3.4 а б в) ключевые признаки, имеющие значения от 1 до 12, так, чтобы деревья стали упорядоченными.



а



б

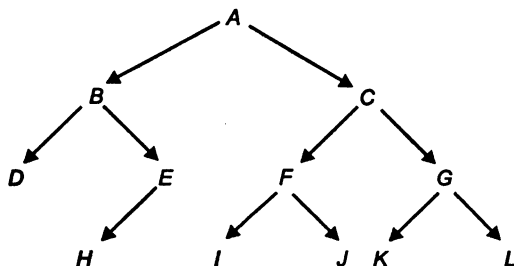


Рис. 3.4. К заданиям 3.56, 3.57

Задание 3.57. В дереве (см. рис. 3.4 в) ключевые признаки изменяются от 1 до 15, вершина F имеет ключ 9. Проставьте остальные ключевые признаки, чтобы обеспечить упорядоченность дерева.

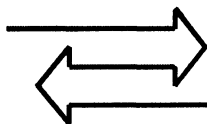
Задание 3.58. Если в бинарном дереве a_2 полных вершин и a_1 неполных, то сколько в нем концевых вершин a_0 ?

Задание 3.59. Сколькими разными способами можно присоединить к бинарному дереву из M записей одну новую запись?

Задание 3.60. Сколько разных конфигураций получается после исключения одной висячей вершины из упорядоченного бинарного дерева, содержащего M записей?

Задание 3.61. Сформулируйте алгоритм перечисления вершин упорядоченного бинарного дерева таким образом, чтобы получилась последовательность записей, отсортированных по возрасту ключа. Как изменится алгоритм, если надо получить массив, упорядоченный по убыванию ключевого признака?

Задание 3.62. Изобразите все возможные бинарные деревья из 4 записей, все деревья порядка 3 из 6 записей, все деревья порядка 4 из 7 записей.



КОМПЛЕКСНЫЕ ЗАДАНИЯ И ТЕСТЫ

4.1.

Комплексные задания

Методические указания

Комплексные задания призваны выявить знания студентов по дисциплине ТЭИС и умение применять эти знания в проектной работе. В процессе выполнения заданий студент должен проявить свои навыки в самостоятельной работе с научно-технической литературой, в обобщении накопленного опыта и умение делать научно обоснованные выводы и рекомендации.

Целями расчета являются:

- закрепление и развитие теоретических знаний, полученных в процессе изучения дисциплины ТЭИС;
- приобретение практических навыков разработки компонент ЭИС;
- выполнение технико-экономического обоснования для выбранных проектных решений.

В качестве объекта исследования может выступать экономическая задача, требующая обработки 3–5 входных документов и получения порядка 3–5 выходных документов или реализации не менее 10 запросов.

Задание должно содержать текстовую и графическую части.

Текстовая часть включает следующие компоненты.

1. Введение. Описываются специфические особенности предметной области, приводится общая характеристика функциональных задач.

2. Разработка требований к ЭИС. Дается постановка задачи, организационная структура объекта управления, основные характеристики информационных потоков, периодичность решения задачи.

3. Проектная часть. Необходима разработка следующих проектных материалов:

- бланков входных и выходных документов;
- структуры всех экономических показателей;
- графа алгоритмических связей показателей;
- модели «сущность – связь»;
- структуры реляционной БД в 3 нормальной форме;
- структуры сетевой двухуровневой БД;
- реализации запросов к реляционной БД средствами реляционной алгебры или конкретной СУБД;
- сравнения реляционной и сетевой БД по критерию минимального объема памяти (длины атрибутов и число записей выбирать самостоятельно).

Задания

Задание 4.1. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Учет движения материалов на складах

А. Входные документы.

A1. Приходный ордер

A2. Лимитная карта

A3. Требование

A4. Товарно-транспортная накладная

В. Выходные документы.

B1. Ведомость прихода на склад

B2. Ведомость расхода по складу

B3. Обратная ведомость по складу

Р е к в и з и т ы :

Дата, Цех-получатель, Номер склада, Код поставщика, Наименование поставщика, Предприятие, Наименование грузоотправителя, Код грузоотправителя, Адрес грузоотправителя, Расчетный счет грузоотправителя, Наименование получателя, Код получателя, Адрес получателя, Расчетный счет получателя, Наименование материала, Код материала, Единица измерения, Цена материала, Норма запаса материала, Количество по документу, Количество принято, Количество затребовано, Остаток материала, Остаток на начало месяца (количество, сумма),

Приход (количество, сумма), Расход (количество, сумма), Остаток на конец месяца (количество, сумма), Количество отпущено, Сумма.

В выходных документах должны быть предусмотрены итоги:

- по документу;
- по складу;
- по кодам поставщиков и получателей;
- по кодам материалов.

Необходима реализация следующих запросов.

- В какие дни объем поставок материала X от поставщика T превышал 200 единиц;
- Какие поставщики поставляют те же материалы, что и поставщик T .

Задание 4.2. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система магазина

Реквизиты:

Номер книжного магазина, Адрес магазина, Фактический товарооборот за прошлый месяц, Дата отчета магазина, Название проданной книги, Автор, Издательство, Цена, Количество проданных книг данного названия на день отчета.

Задание 4.3. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система банка

1. Идентификационный номер клиента ИНК является уникальным.

2. Номера могут присваиваться счетам в любом отделении банка, они являются уникальными. Номер счета однозначно определяет тип счета, отделение и имя клиента.

3. Номера кассиров являются уникальными.

4. Кассир может работать в разных отделениях, однако в течение дня он не покидает данное отделение.

5. Номер ссуды отличается от номера счета. Счет и ссуда являются разными объектами.

6. Клиент может иметь несколько счетов с разными номерами в одном отделении банка.

7. В каждом отделении имеется только один управляющий.

А. Входные документы.

A1. Внос/изъятие денег.

В. Выходные документы.

B1. Отчет управляющего отделением.

B2. Состояние ссуд по отделению.

B3. Отчет кассира.

Р е к в и з и т ы:

Номер отделения, Название отделения, Адрес отделения, Имя управляющего, Номер счета, Тип счета, ИНК, Имя клиента, Адрес клиента, Телефон клиента, Дата и время операции, Тип операции, Номер операции, Сумма, Номер ссуды, Назначено по ссуде, Получено по ссуде, Недополучено по ссуде, Процент, Номер кассира, Имя кассира, Дата, Наличность на начало дня, Наличность на конец дня, Принято наличными, Выдано наличными, Остаток, Дата рождения клиента.

Необходима реализация следующих запросов:

- список клиентов, у которых остаток по счету превышает 3000 руб.;
- в какие дни сумма выданных денег превышает сумму принятых денег.

Задание 4.4. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система для университета

1. Каждому студенту назначается куратор.
2. В семестре студент изучает несколько курсов.
3. Один и тот же курс могут вести разные преподаватели в разных семестрах.

4. На данный семестр студенты объединяются в группу.

А. Входные документы.

A1. Расписание занятий студента на семестр.

A2. Список студентов, слушающих курс.

A3. Список курсов.

В. Выходные документы.

B1. Сведения о студенте на конец семестра.

В2. Сведения о выпускниках прошлых лет.

В3. Сведения о выпуске студентов кафедрами.

Реквизиты:

Семестр, Дата начала семестра, Дата окончания семестра, Номер студента, Имя студента, Куратор, Номер курса, Название курса, Имя преподавателя, День и время, Номер аудитории, Кафедра, Оценка, Адрес студента, Присвоенная степень, Дата присвоения степени, Должность, Место работы, Телефон, Всего студентов кафедры.

Задание 4.5. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система футбольной команды

Реквизиты:

Стаж тренера, Позиция тренера, Поражение, Победа, Награды-тренера, Число-зрителей, Дата-игры, Результат-хозяев-поля, Команда-хозяев-поля, Стадион, Результат-гостей, Команда-гостей, Победитель, Счет, Общее-число-нарушений, Имя-игрока, Рост-игрока, Стаж-игрока, Вес-игрока, Основная-позиция-игрока, Дата-рождения-игрока, Класс-игрока-в-команде, Количество-травм-игрока, Игровое-время-игрока, Число-нарушений-игрока, Цвет-формы, Название-команды, Награды-команды.

Задание 4.6. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система морских перевозок

Реквизиты:

Дата, Наименование судна, Скорость хода, Грузоподъемность, Дата начала задания, Порт окончания задания, Вид валюты, Сумма, Доход за задание, Расход за задание, Имя арендатора, Дата прибытия в порт, Дата отправления из порта, Наименование порта захода, Код груза, Операция над грузом, Операция в порту, Вес груза, Имя грузовладельца, Дата готовности груза к перевозке, Дата доставки груза получателю, Порт отправления груза, Порт назначения груза.

Задание 4.7. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система страховой компании

Реквизиты:

ФИО служащего (агента), Номер служащего, Отдел, Руководитель отдела, Зарплата, Номер страхового полиса, Агент, оформивший полис, ФИО получателя страховки, Код события страхования, Дата оформления полиса, Срок действия полиса, Периодичность страховых взносов, Сумма страховых взносов, Сумма страховой выплаты.

Задание 4.8. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система школы

Реквизиты:

ФИО учителя, Номер учителя, Преподаваемая дисциплина, Номер класса, Дата, Количество уроков за день, Квалификационный разряд учителя, Оплата за урок, Месяц, Начислено учителю за месяц, Отчисления в пенсионный фонд, Сумма налогов, Другие удержания, Выдано на руки.

Задание 4.9. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система автостанции

Реквизиты:

Номер рейса, Пункт отправления, Пункт назначения, Дата отправления, Номер автобуса, Основной водитель, Сменный водитель, Количество мест, Проданные места на момент отправления, ФИО водителя, Номер водителя, Дата, Время в пути, Регистрационный номер проданного билета, Номер рейса, Дата отправления, Пункт назначения, Стоимость билета.

Задание 4.10. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система торговой фирмы

Реквизиты:

Наименование товара, Код товара, Единица измерения, Цена, Норма запаса, Фио торгового агента, Номер торгового агента, Номер заказа, Дата оформления заказа, Дата исполнения заказа, Количество заказанного товара, Фио клиента, Адрес и телефон клиента, Сумма по заказу, Сумма предоплаты.

Задание 4.11. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система приемной комиссии вуза

Реквизиты:

ФИО абитуриента, Код абитуриента, Год рождения, Пол, Суммарный балл экзаменов, Код факультета, Название факультета, ФИО декана, Количество мест, Проходной балл, Название экзаменационной дисциплины, Дата сдачи экзамена, Оценка экзамена, Код группы, ФИО зачисленного абитуриента.

Задание 4.12. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система научно-исследовательского института (НИИ)

Реквизиты:

Название НИИ, Директор НИИ, Код отдела, Число сотрудников отдела, Код темы, Дата начала темы, Дата окончания темы, Приоритет темы, Заказчик темы, Объем финансирования темы заказчиком, Код работы в теме, Продолжительность работы, ФИО исполнителя работы.

Задание 4.13. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система кинотеатра

Какие новые фильмы вышли в данном месяце. Выдать информацию о режиссере, исполнителях главной мужской и женской

ролей. В каком кинотеатре фильм демонстрируется, есть ли билеты в кассах кинотеатра, какова цена билета.

Список реквизитов выберите самостоятельно.

Задание 4.14. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система музыкальной радиостанции

Выдать список песен и их исполнителей на любой из часов текущего дня. Кто и на какую сумму заказал рекламу в текущем месяце. Сколько было заказов радиослушателей на определенную песню за день.

Список реквизитов выберите самостоятельно.

Задание 4.15. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Учет выработки в цехе предприятия

Реквизиты:

Номер рапорта, Месяц, Год, Код цеха, Название цеха, Код участка, Название участка, Код бригады, ФИО бригадира, Вид оплаты, Код производственных затрат, Номер детали, Дата, Количество деталей, выданных в работу, Количество деталей, принятых, годных, Количество деталей, бракованных, Количество деталей, бракованных, оплаченных.

Необходимы итоги по всем количественным характеристикам для каждого участка и цеха.

Задание 4.16. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Расчет заработной платы

Входные документы.

- A1. Справочник работающих.
- A2. Табель учета рабочего времени.
- A3. Больничные листы.

Выходные документы.

В1. Расчетный листок.

В2. Ведомость на выдачу аванса.

В3. Ведомость на выдачу заработной платы.

Реквизиты:

Участок, Табельный номер, ФИО, Код профессии, Система оплаты, Часовая тарифная ставка, Плановый аванс, Код налогоплательщика, Месяц, Год, Всего отработано дней, Отработанные часы, Число дней, оплачиваемых по больничному листу, Пособие в процентах к зарплате, Сумма пособия, Начисленная зарплата, Удержания – налоги, Удержания – пенсионный фонд, Сумма аванса, Сумма за вторую половину месяца.

Задание 4.17. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Поставка продукции на экспорт

(ВТО – внешнеторговое объединение).

Реквизиты:

Код ВТО-отправителя, Название ВТО-отправителя, Название страны-получателя, Код продукции, Наименование продукции, Код вида исполнения, Цена продукции, Номер заказа, Дата, Количество продукции, Дата отгрузки, Сумма заказа, Объем поставок по стране, Объем поставок по ВТО.

Задание 4.18. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система аукционных торгов

Изделиям, выставляемым на аукцион, выделяются номера лоты и определяется их стартовая цена. Лица и организации подают заявки на участие в аукционе с уплатой вступительного взноса, указанием списка лотов, в торгах по которым предполагается принять участие, и перечислением на ответственное хранение 10% суммы стартовой цены указанных лотов.

Реквизиты:

Название изделия, Источник поступления, Стартовая цена, Номер участника аукциона, Название участника, Размер вступи-

тельного взноса, Сумма предоплаты за лоты, Цена продажи лота, Покупатель лота.

Необходимо предусмотреть выдачу документов:

- список проданных лотов;
- список лотов, снятых с торгов;
- список покупок для участников торгов;
- ведомость возврата взносов и предоплат.

Задание 4.19. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Операции с облигациями

Реквизиты облигации: серия, выпуск, дата размещения, дата погашения, номинал, средняя цена, доходность, объем.

Реквизиты дилеров: наименование организации, номер лицензии, адрес, телефон.

Реквизиты покупателей и продавцов: наименование организации, адрес, телефон.

Реквизиты покупок и продаж: дата, цена покупки/продажи, куплено/продано, комиссия за покупку/ продажу.

Задание 4.20. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Выпуск в обращение ценных бумаг

Банк выпускает документ, на основе которого проводится эмиссия ценных бумаг. На каждый вид ценных бумаг имеется комплект документов, регламентирующих выпуск и условия обращения. Кроме того, существует ряд справочных систем и коммерческих баз данных, предоставляющих информацию по обороту ценных бумаг.

Реквизиты:

Название банка, Номер лицензии, Адрес, Телефон/факс, Размер уставного фонда, Процент уставного фонда на эмиссию, Название ценной бумаги, Банк-эмитент, Номер выпуска, Объем эмиссии, Название документа, регламентирующего выпуск, код документа в Министерстве юстиции, Дата принятия, Срок действия, Первичная цена ценной бумаги, Процент, выплачиваемый

по ценной бумаге, Дата регистрации текущей цены в системе, Текущая цена ценной бумаги, Название системы/базы данных, Цена за услуги системы/базы данных, Количество документов в системе на дату регистрации.

Задание 4.21. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система музея

Реквизиты:

Регистрационный номер экспоната, Название экспоната, Автор экспоната, Дата поступления, Дата выбытия, Дата реставрации, ФИО реставратора, Оценочная стоимость экспоната, Название выставки, Период проведения выставки, ФИО владельца экспоната.

Экспонаты переходят от музея к музею, реставрируются и участвуют в выставках.

Задание 4.22. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Задача учета рабочего времени

Реквизиты:

№ табеля, Месяц, Год, Код цеха, Наименование цеха, Код участка, ФИО рабочего, Табельный номер рабочего, Категория рабочего, Профессия рабочего, Разряд работы, Вид оплат, Код производственных затрат, Дата, Отработано часов на дату, Отработано часов за месяц, Отработано дней за месяц, Отработано часов по видам оплат.

Задание 4.23. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Учет коммунальных платежей

Реквизиты:

ФИО квартиросъемщика, Адрес, Платежные реквизиты по оплате жилья, Платежные реквизиты по оплате электроэнергии, платежные реквизиты по оплате газа, Платежные реквизиты по оп-

лате телефона, Ставка оплаты жилья, Ставка оплаты электроэнергии, Ставка оплаты газа, Ставка оплаты телефона, Льготы по оплате жилья, Льготы по оплате электроэнергии, Льготы по оплате газа, Льготы по оплате телефона, Номер квартиросъемщика при оплате жилья, Номер квартиросъемщика при оплате электроэнергии, Номер квартиросъемщика при оплате газа, Номер квартиросъемщика при оплате телефона, Дата оплаты, Сумма оплаты жилья, Сумма оплаты электроэнергии, Сумма оплаты газа, Сумма оплаты телефона.

Задание 4.24. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Задача отпуска материала со склада

Реквизиты:

Код склада, Код склада получателя, Наименование материала, Код материала, Код единицы измерения, Цена материала, Код производственных затрат, Код операции, Номер требования, Дата выдачи, Количество отпущено, Расход материала по складу, Расход материала по цеху.

Задание 4.25. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система аптеки

Реквизиты:

Номер аптеки, Телефон, Адрес, Название лекарства, Страна-производитель, Норма запаса лекарства на дату, Дата, Номер льготного рецепта, Сумма по льготному лекарству, Сумма по всем лекарствам на дату.

Задание 4.26. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Задача учета производственного травматизма

Реквизиты:

Номер акта, Предприятие, ФИО пострадавшего, Возраст, Должность, Вид происшествия, Причина несчастного случая,

Дата несчастного случая, Продолжительность нетрудоспособности, Выплачено по нетрудоспособности, Стоимость испорченного оборудования и инструмента, Общая сумма ущерба.

Необходимы сводки о количестве пострадавших и числе дней нетрудоспособности в течение года по предприятию.

Задание 4.27. Выполните комплексное задание для приведенных ниже исходных данных. Список реквизитов можно видоизменять. Все обозначения выберите самостоятельно.

Информационная система фотовыставки

Реквизиты:

Название фотовыставки, Период проведения выставки, ФИО автора, Адрес, Телефон, Название экспоната, Дата поступления экспоната, Балл экспоната на выставке.

Экпонат данного автора может участвовать в нескольких выставках.

4.2.

Список контрольных вопросов по теории экономических информационных систем

1. Охарактеризуйте понятия «информация», «данные», «экономическая информация», их основные свойства и область применения.

2. Охарактеризуйте понятие «тезаурус», взаимосвязи понятий «информация» и «тезаурус», приведите примеры тезаурусов.

3. Как связано понятие «тезаурус» с передачей и приемом информации?

4. Охарактеризуйте основные понятия, связанные с передачей информации: «источник», «приемник», «носитель».

5. Перечислите основные компоненты, назначение и основные функции экономической информационной системы.

6. Назовите основные компоненты информационного процесса.

7. Что такое информационная база? Приведите примеры информационных баз.

8. Дайте определение понятий «информационная потребность» и «релевантность».

9. Дайте определение понятия «информационная услуга». Охарактеризуйте современный рынок информационных услуг.

10. Дайте определение и объясните на примере понятие «синтаксис».

11. Дайте определение и объясните на примере понятие «семантика».

12. Дайте определение и объясните на примере понятие «носитель данных».

13. Приведите примеры машинных и немашинных носителей данных.

14. Назовите этапы жизненного цикла экономической информационной системы.

15. Что образует немашинную информационную базу?

16. На какие виды разделяется экономическая информация по функциям управления?

17. Что включает плановая (директивная) информация?

18. Что включает учетная информация?

19. Что включает нормативно-справочная информация?

20. Что включает отчетно-статистическая информация?

21. Как классифицируются информационные системы в зависимости от степени (уровня) автоматизации?

22. Как классифицируются информационные системы по сфере применения?

23. В каких основных режимах могут работать системы обработки данных?

24. Охарактеризуйте понятие «пользователь информационной системы», приведите классификацию пользователей.

25. Дайте определение понятия «экономический документ». Чем отличаются текстовые (в свободном формате) и форматированные документы?

26. Назовите компоненты экономического документа.

27. Как выполняется нормализация документа, каковы свойства нормализованного массива?

28. Охарактеризуйте основные единицы информации – реквизит и составную единицу информации (СЕИ). Как организована иерархия единиц информации?

29. Назовите свойства и типы реквизитов.

30. Дайте определение понятий «имя реквизита» и «множество значений» (домен) реквизита.

31. Дайте определение понятия «показатель».

32. Перечислите свойства нормализованного массива данных и следствия из них.

33. Охарактеризуйте понятия «схема», «экземпляр», «первичный ключ» и «вторичный ключ».

34. Назовите свойства кодовых обозначений, приведите примеры систем кодирования.

35. Охарактеризуйте понятие «функциональная зависимость».

36. Можно ли рассматривать понятие «отношение» как модель понятия «нормализованный массив»?

37. Назовите компоненты модели данных.

38. Какие информационные конструкции, операции и ограничения характеризуют реляционную модель данных?

39. Укажите определение и основные свойства операции проекции.

40. Как связана операция группировки с понятиями нормализованного файла и проекции?

41. Определите естественное соединение как частный случай операции соединения.

42. Охарактеризуйте эффект ловушки связей и сформулируйте свойство соединения без потерь.

43. Назовите разновидности операции корректировки.

44. Перечислите основные свойства языка SQL.

45. Как реализуются операции поиска и проекции средствами языка SQL?

46. Запишите операцию естественного соединения средствами языка SQL.

47. Дайте определения неполных и транзитивных функциональных зависимостей.

48. Является ли вторая нормальная форма частным случаем третьей нормальной формы?

49. В каком случае декомпозиция отношения обладает свойством соединения без потерь?

50. Назовите минимально необходимый список теорем о функциональных зависимостях.

51. Приведите примеры минимальных и неминимальных покрытий функциональных зависимостей.

52. С какой целью при построении ЗНФ отношений создается при необходимости отношение из реквизитов первичного ключа?

53. Определите понятия «схема» и «экземпляр» для иерархической базы данных, а также понятие «иерархическая упорядоченность».

54. Приведите преимущества иерархической модели данных перед реляционной.

55. Сформулируйте определение сетевой схемы и основные элементы сетевой схемы.

56. Укажите численные характеристики операции поиска.

57. Охарактеризуйте прямой и последовательный доступ к данным в информационном фонде.

58. Сформулируйте критерий выбора прямого или последовательного метода доступа.

59. Что понимается под базой данных?

60. Что входит в систему классификации и кодирования?

61. Что такое классификатор?

62. Что такое предметная область?

63. Охарактеризуйте понятия «экономический эффект» и «экономическая эффективность».

64. Укажите основные показатели экономической эффективности информационных систем.

65. Какую роль играет понятие «модель данных» в системе управления базой данных?

66. Охарактеризуйте способы размещения данных на машинном носителе, понятия «адрес», «доступ», «указатель».

67. Назовите известные вам методы сортировки. В чем необходимость сортировки?

68. Что такое прямой доступ?

69. Что такое последовательный доступ?

70. Что отражает оперативность информации?

71. Что понимается под функциональными компонентами экономической информационной системы?

72. Что включают компоненты системы обработки данных?

73. Какие подсистемы хозяйствующего субъекта рассматриваются в теории экономических информационных систем?

74. Что определяет синтаксический анализ экономической информации?

75. Что позволяет установить семантический анализ экономической информации?

76. С какой целью проводится прагматический анализ экономической информации?

77. Что называется *доменом* и *кортежем* реляционной таблицы-отношения?

78. Что понимается под точностью информации?

79. Что понимается под достоверностью информации?

80. Охарактеризуйте и дайте описание элементов языка описания данных и языка манипулирования данными системы управления базой данных Access.

81. Из каких реквизитов состоит первичный ключ отношения, в котором нет справедливых функциональных зависимостей?

82. Назовите способы определения объема экономической информации.

83. Перечислите преимущества и недостатки последовательной организации данных.

84. Дайте определение инвертированного массива.

85. Приведите сравнительную оценку методов поиска для последовательной организации данных.

86. Можно ли определить дерево как организацию данных, в которой каждый элемент, кроме корня, имеет только один предшествующий элемент, и наличие корня обязательно?

4.3. Тесты

1. Записью называется:

а) элемент файла;

б) набор реквизитов;

в) элементарная единица информации?

2. Отношение $R(A, B)$, где A, B – реквизиты, имеет максимум:

а) первую нормальную форму;

б) вторую нормальную форму;

в) третью нормальную форму?

3. По результатам измерений среднее время выполнения одного запроса на ЭВМ равно 2 мин., интенсивность потока запросов равна 20 запросам в час. Коэффициент использования ЭВМ равен:

а) $4/5$;

б) $3/4$;

в) $2/3$?

4. Доля выборки записей при одном запросе к файлу из 10 тыс. записей составляет в среднем 40%. Файл должен быть организован:

- а) прямым методом;
- б) индексным методом;
- в) последовательным методом?

5. Какой метод организации данных позволяет быстрее всего реализовать операцию корректировки записей:

- а) древовидный;
- б) цепной;
- в) последовательный?

6. Какой метод организации данных использует меньший объем дополнительной памяти:

- а) последовательный с индексом для каждой записи;
- б) цепной;
- в) древовидный?

7. Какие записи называются синонимами:

- а) записи с одинаковым значением адресной функции;
- б) записи с одинаковым значением ключа;
- в) записи с одинаковыми значениями всех реквизитов?

8. Может ли указанная ниже иерархическая база данных содержать информацию о человеке, который трудится индивидуально, вне бригады?

Корневое отношение – БРИГАДА.

Зависимое отношение – РАБОЧИЙ.

- а) да;
 - б) нет;
 - в) да, если такой случай единственный в базе данных?
9. Даны отношения R и S .

R	A	B
	8	6
	1	2
	2	2
	1	1

S	C	D
	1	2
	3	2
	4	3
	4	1

Сколько строк содержит результат соединения R и S по условию $A <> C$:

- а) 8;
- б) 10;
- в) 14?

10. Какой процесс обработки данных выполняется за время, пропорциональное $M \cdot \log(M)$:

- а) сортировка;
- б) поиск;
- в) корректировка?

11. Даны функциональные зависимости $A \rightarrow B, C$; $B \rightarrow D$; $C \rightarrow E$.

Какая из функциональных зависимостей является их следствием:

- а) $A \rightarrow D$;
- б) $B \rightarrow C$;
- в) $B \rightarrow E$?

12. Отношение $R(A, B, C)$ имеет ключ A, B . Отношение $S(B, D)$ имеет ключ B . Результат соединения отношений R и S по условию $B=B$ имеет ключ:

- а) B ;
- б) A, B ;
- в) A, B, C, D ?

13. Укажите буквы, соответствующие правильным утверждениям:

а) двухуровневая сетевая структура базы данных может содержать циклические компоненты;

б) любая функциональная зависимость в реляционной базе данных выводима из минимального покрытия этих функциональных зависимостей;

в) если в отношении нет функциональных зависимостей, то в нем нет вероятного ключа.

14. Даны отношения в третьей нормальной форме:

$R(A, B, C)$ с ключом A ;

$S(D, E)$ с ключом D ;

$T(A, D, F, \text{ДАТА})$ с ключом A, D, F .

В отношение T добавлен реквизит ГОД. Третья нормальная форма отношений R, S, T :

- а) сохранится;
- б) исчезнет;

в) ответ зависит от наличия функциональных зависимостей $A \rightarrow \text{ГОД}$ и $D \rightarrow \text{ГОД}$.

15. Даны отношения $R(A, B)$; $T(B, C)$; $Q(A, C)$. Знаком «*» обозначена операция соединения отношений по равенству реквизитов.

Равенство $(R * T) * Q = R * (T * Q)$:

а) всегда соблюдается;

б) никогда не соблюдается;

в) ответ зависит от смысла реквизитов A , B , C .

16. Какая модель данных разрешает участие приведенного отношения в качестве основного в одном веерном отношении и одновременно в качестве зависимого – в другом:

а) иерархическая;

б) двухуровневая сетевая;

в) реляционная.

17. Дано отношение с реквизитами Завод, Продукция, Цена, Дата, Выпуск. Первичным ключом будет набор реквизитов:

1) Продукция, Цена, Дата;

2) Завод, Дата;

3) Завод, Продукция, Дата;

4) Завод, Продукция;

5) Завод, Продукция, Выпуск.

18. Дано отношение с реквизитами Студент, Дисциплина, Дата, Преподаватель, Оценка. Первичным ключом будет набор реквизитов:

1) Дисциплина, Дата;

2) Студент, Дисциплина, Дата;

3) Дисциплина, Преподаватель;

4) Студент, Преподаватель, Оценка;

5) Студент, Дисциплина, Оценка.

19. Дано отношение с реквизитами Рабочий, Деталь, Вес_детали, Дата, Количество. Первичным ключом будет набор реквизитов:

1) Деталь, Дата, Количество;

2) Рабочий, Деталь;

3) Рабочий, Деталь, Дата;

4) Деталь, Дата;

5) Рабочий, Деталь, Количество.

20. Отрицанием запроса к базе данных

**ДОЛЖНОСТЬ = «ИНЖЕНЕР» И ЗАРПЛАТА > 6000 ИЛИ
СТАЖ > 20**

является запрос:

- 1) Должность ≠ «инженер» или Зарплата ≤ 6000 и Стаж ≤ 20;
- 2) Должность ≠ «инженер» и Зарплата ≤ 6000 или Стаж ≤ 20;
- 3) Должность ≠ «инженер» или Зарплата ≤ 6000 и Стаж ≤ 20;
- 4) Должность ≠ «инженер» или Зарплата ≤ 6000 и Стаж ≤ 20;
- 5) Должность ≠ «инженер» и Зарплата ≤ 6000 и Стаж ≤ 20.

21. Переместившееся на максимальное расстояние значение в столбце «ВЕС» в базе данных

Номер записи	ФИО	Вес
1	Иванов	67
2	Ликин	75
3	Сидоров	73
4	Асеев	74
5	Сизов	68
6	Сытин	66

после сортировки таблицы по реквизиту «ФИО» в порядке возрастания, равно:

- 1) 66; 2) 67; 3) 74; 4) 75; 5) 68.

22. Отрицанием запроса к базе данных

(МАРКА = «АУДИ» ИЛИ ЦЕНА > 19 000) И ЦВЕТ = «ЧЕРНЫЙ»

является запрос:

- 1) Марка ≠ «Ауди» или Цена ≤ 19 000 и Цвет ≠ «черный»;
- 2) Марка ≠ «Ауди» и Цена ≤ 19 000 или Цвет ≠ «черный»;
- 3) Марка ≠ «Ауди» и Цена > 19 000 или Цвет = «черный»;
- 4) Марка ≠ «Ауди» или Цена ≤ 19 000 и Цвет = «черный»;
- 5) Марка ≠ «Ауди» и Цена > 19 000 и Цвет ≠ «черный».

23. Операциями над структурой документа являются:

- 1) нормализация;
- 2) декомпозиция;
- 3) инверсия;
- 4) свертка;
- 5) композиция.

24. Сколько показателей в базе данных с реквизитами:

Дата, Наименование судна, Скорость хода, Грузоподъемность, Дата начала задания, Порт окончания задания, Вид валюты, Сумма, Доход за задание, Расход за задание, Имя арендато-

ра, Дата прибытия в порт, Дата отправления из порта, Наименование порта захода, Код груза, Операция над грузом, Операция в порту, Вес груза, Имя грузовладельца, Дата готовности груза к перевозке, Дата доставки груза получателю, Порт отправления груза, Порт назначения груза.

- а) 3;
- б) 4;
- в) 5;
- г) 6?

25. Декомпозиция показателя на несколько показателей:

- 1) всегда возможна;
- 2) невозможна;
- 3) ответ зависит от смысла атрибутов, входящих в показатель.

26. Даны следующие утверждения.

- 1) Структуру базы данных ЭИС можно представить как структуру одной составной единицы информации;
- 2) Атрибут не имеет структуры, потому что его значение не разделяется на поименованные части;
- 3) Ролевые атрибуты могут иметь разные области определения.

Правильными утверждениями являются:

- а) 1;
- б) 2;
- в) 3;
- г) 1 и 2;
- д) 1 и 3;
- е) 2 и 3;
- ж) все утверждения.

27. Даны три базы данных:

БД1(отдел, телефон, ФИО сотрудника, тема работы);
БД2(аптека, телефон, лекарство, страна-изготовитель);
БД3(судно, грузоподъемность, порт приписки, судовладелец).
Не содержат атрибутов-оснований:

- а) БД1;
- б) БД2;
- в) БД3;
- г) БД1 и БД2;
- д) БД1 и БД3;
- е) БД2 и БД3;
- ж) все базы данных.

28. Суммирование итогов является корректным:

1) по атрибуту, полученному в результате выборки по всей базе данных;

2) по атрибуту-основанию, расположенному в файле;

3) по атрибуту-основанию, расположенному в показателе?

29. Не имеет экономического смысла:

1) база данных, содержащая только атрибуты-признаки;

2) база данных, содержащая только атрибуты-основания;

3) база данных, содержащая только атрибуты с логическими значениями.

30. Отдельное сообщение:

1) обязательно должно иметь форму экономического показателя;

2) не обязательно должно иметь форму экономического показателя;

3) обязательно должно быть форматированным.

31. Операциями над значениями составных единиц информации являются:

1) сортировка;

2) выборка;

3) транспонирование;

4) корректировка.

32. Сколько реквизитов не входят ни в один показатель в базе данных с реквизитами:

Название объекта, Год изготовления, Инвентарный номер, Дата ликвидации, Первоначальная стоимость, Сумма износа, Остаточная стоимость.

а) 0;

б) 1;

в) 2;

г) 3?

33. Даны следующие утверждения:

1) Существует эквивалентное преобразование двухуровневой сети в реляционное представление;

2) Существует эквивалентное преобразование двухуровневой сети в одну иерархическую базу данных;

3) Существует эквивалентное преобразование многоуровневой сети в одну иерархическую базу данных.

Правильными утверждениями являются:

а) 1;

- б) 2;
- в) 3;
- г) 1 и 2;
- д) 1 и 3;
- е) 2 и 3;
- ж) все утверждения.

34. Отношения $T*S$ и $S*T$:

- 1) равны как множества;
- 2) не равны как множества;
- 3) содержат разное количество строк.

35. Пары связей, показанные ниже,

(ИЗДЕЛИЕ, КОЛЕСО);

(ИЗДЕЛИЕ, АВТОМОБИЛЬ);

(ТРАНСПОРТ, АВТОМОБИЛЬ);

(ТРАНСПОРТ, САМОЛЕТ),

представляют собой:

- а) структуру сущностей и связей;
- б) структуру событий;
- в) структуру понятий.

36. Какая модель представления знаний реализует наследование свойств:

- а) фреймовая модель;
- б) модель семантической сети;
- в) реляционная модель?

37. Какая модель представления знаний использует присоединенные процедуры:

- а) продукционная модель;
- б) фреймовая модель;
- в) модель семантической сети?

38. Связь между понятиями «Инженер» и «Специалист» имеет вид:

- а) есть некоторый;
- б) есть часть;
- в) другой вид.

39. Связь между понятиями «Отдел» и «Учреждение» имеет вид:

- а) есть некоторый;
- б) есть часть;
- в) другой вид.

40. В базе данных с реквизитами:

Регистрационный номер экспоната, Название экспоната, Автор экспоната, Дата поступления, Дата выбытия, Дата реставрации, ФИО реставратора, Оценочная стоимость экспоната, Название выставки, Период проведения выставки, ФИО владельца экспоната,

реквизит «Автор экспоната» представляет:

- а) объект;
- б) свойство объекта;
- в) взаимодействие объектов;
- г) свойство взаимодействия?

41. В базе данных с реквизитами:

Наименование товара, Код товара, Единица измерения, Цена, Норма запаса, ФИО торгового агента, Номер торгового агента, Номер заказа, Дата оформления заказа, Дата исполнения заказа, Количество заказанного товара, ФИО клиента, Адрес и телефон клиента, Сумма по заказу, Сумма предоплаты,

реквизит «Сумма предоплаты» представляет:

- а) объект;
- б) свойство объекта;
- в) взаимодействие объектов;
- г) свойство взаимодействия?

42. В модели сущностей и связей дуга может соединять:

- а) сущность и сущность;
- б) связь и связь;
- в) сущность и связь?

43. Существует база данных с реквизитами:

ФИО абитуриента, Код абитуриента, Год рождения, Пол, Суммарный балл экзаменов, Код факультета, Название факультета, ФИО декана, Количество мест, Проходной балл, Название экзаменационной дисциплины, Дата сдачи экзамена, Оценка экзамена, Код группы, ФИО зачисленного абитуриента.

Сколько реквизитов представляют взаимодействие объектов:

- а) 0;
- б) 2;
- в) 3;
- г) 4?

44. Существует база данных с реквизитами:

Наименование товара, Код товара, Единица измерения, Цена, Норма запаса, ФИО торгового агента, Номер торгового агента, Номер заказа, Дата оформления заказа, Дата исполнения заказа,

Количество заказанного товара, ФИО клиента, Адрес и телефон клиента, Сумма по заказу, Сумма предоплаты.

Сколько реквизитов представляют свойство взаимодействия:

- а) 0;
- б) 5;
- в) 3;
- г) 4?

45. Правильным утверждением является:

а) в дереве каждый элемент имеет только один предшествующий элемент;

б) дерево соответствует связному графу, не содержащему циклов;

в) неверно, что число записей в дереве всегда меньше, чем произведение порядка дерева на его ранг.

46. В бинарном дереве на рис. 3.4 б для записи D соседней записью является:

- а) F ;
- б) G ;
- в) J ?

47. Даны следующие утверждения:

1) перед включением новой записи необходим поиск ее местоположения относительно существующих записей;

2) все методы сортировки основаны на сравнении пар ключевых признаков записей;

3) в упорядоченном бинарном дереве, построенном на основе массива, упорядоченного по невозрастанию ключевых признаков, заполнены только левые адреса связи.

Правильными утверждениями не являются:

- а) 1;
- б) 2;
- в) 3;
- г) 1 и 2;
- д) 1 и 3;
- е) 2 и 3;

ж) все утверждения.

48. Вероятность того, что массив из M записей случайно окажется упорядоченным, равна:

- а) $1 / M$;
- б) $1 / (M!)$;
- в) $2 / (M!)$.

49. Укажите буквы, соответствующие правильным утверждениям:

а) двухуровневая сетевая структура базы данных может содержать циклические компоненты;

б) любая функциональная зависимость для базы данных является следствием из минимального покрытия этих функциональных зависимостей;

в) если в отношении нет функциональных зависимостей, то в нем нет вероятного ключа.

50. Для отношения $R(A, B, C)$ дан запрос « $A = 10$ или $A <> 10$ и $B <> 2$ ». Эквивалентным для него запросом является:

а) «не ($A = 10$ или $B = 2$)»;

б) «не ($A = 10$ и $B = 2$)»;

в) « $A = 10$ или $B <> 2$ »;

г) « $B <> 2$ »?

51. Сколько вероятных ключей в отношении:

T1	KOS	KOGR	WKII	DUBL	PRIN
	AK	61	WA	570	WS
	AT	44	PR	751	WA
	AK	20	AI	856	RI
	AU	73	WA	536	TX
	AT	15	PR	751	AX
	AK	94	FY	456	PI
	AU	30	CM	846	CI
	AT	52	VD	751	WS

1) 2;

2) 3;

3) 4;

4) 1?

52. Составной единицей информации не является:

1) отношение;

2) байт;

3) показатель.

53. Операцией реляционной алгебры не является:

1) проекция;

2) соединение;

3) инверсия;

4) выборка.

54. Дано отношение R .

R	A	B	C	X
	$a1$	$b1$	$c1$	$x1$
	$a1$	$b2$	$c1$	$x1$
	$a1$	$b2$	$c2$	$x1$
	$a2$	$b1$	$c1$	$x1$
	$a2$	$b2$	$c1$	$x2$
	$a2$	$b2$	$c2$	$x2$
	$a3$	$b2$	$c1$	$x3$
	$a3$	$b2$	$c2$	$x3$

В отношении R соблюдается функциональная зависимость $A, B, C \rightarrow X$. В проекции $T = R[B, C, X]$ соблюдается функциональная зависимость:

- 1) $B, C \rightarrow X$;
- 2) $B \rightarrow X$;
- 3) никакая.

55. Видовым понятием по отношению к родовому понятию «Рабочий» является понятие:

- 1) «Слесарь»;
- 2) «Технолог»;
- 3) «Конструктор»;
- 4) «Литейщик».

56. Даны отношения:

$T1$ (Цех, Код изделия, План 2003 года);

$T2$ (Код материала, Цена материала);

$T3$ (Код изделия, Цена изделия);

$T4$ (Код материала, Код изделия, Норма расхода).

Количество верных отношений, которые будут установлены при формировании двухуровневой сетевой базы данных, равно:

- а) 1;
- б) 2;
- в) 3;
- г) 4.

57. Даны отношения:

$T1$ (Музей, Город);

$T2$ (Музей, Экспонат, Год поступления);

$T3$ (Экспонат, Реставратор, Год реставрации);

$T4$ (Экспонат, Оценочная стоимость).

Количество верных отношений, которые будут установлены при формировании двухуровневой сетевой базы данных, равно:

- а) 1;
- б) 2;
- в) 3;
- г) 4.

58. Даны следующие утверждения:

1) если в отношении продублировать часть строк, то оно будет представлять то же множество сообщений, что и до преобразования;

2) если в отношении продублировать часть столбцов, то оно будет представлять то же множество сообщений, что и до преобразования;

3) если в отношении заменить текстовые значения объектов их цифровыми кодами, то оно будет представлять то же множество сообщений, что и до преобразования.

Правильными утверждениями являются:

- а) 1;
- б) 2;
- в) 3;
- г) 1 и 2;
- д) 1 и 3;
- е) 2 и 3;
- ж) все утверждения.

Рекомендуемая литература

1. Автоматизированные информационные технологии в экономике: Учебник / Под ред. Г.А. Титоренко. – М.: Компьютер, ЮНИТИ, 1998.
2. Гудман С., Хидеттнеми С. Введение в разработку и анализ алгоритмов: Пер. с англ. – М.: Мир, 1981.
3. Дейт К. Введение в системы баз данных. – 6-е изд.: Пер. с англ. К.; М.; СПб.: Издательский дом «Вильямс», 1999.
4. Евдокимов В.В. и др. Экономическая информатика: Учебник для вузов / Под ред. В.В. Евдокимова. – СПб.: Питер, 1997.
5. Информационные системы в экономике : Под ред. В.В. Дика. – М.: Финансы и статистика, 1996.
6. Мейер Д. Теория реляционных баз данных: Пер. с англ. – М.: Мир, 1987.
7. Мишенин А.И. Теория экономических информационных систем. Учебник. – 4-е изд., доп. и перераб. – М.: Финансы и статистика, 2003.
8. Подольский В.И., Дик В.В., Уринцов А.И. Информационные системы бухгалтерского учета / Под ред. В.И. Подольского. – М.: Аудит, Юнити, 1998.
9. Салмин С.П. Управленческие информационные комплексы и автоматизированные информационные технологии. – Н. Новгород: Изд-во ННГУ, 2000.
10. Сборник задач по курсу «Теория экономических информационных систем» / Сост. С.П. Салмин. – Н. Новгород: НКИ, 2001.
11. Теория экономических информационных систем: Курс лекций / Сост. А.М. Сидоренко. – Н. Новгород: НКИ, 1999.
12. Ульман Дж. Основы систем баз данных: Пер. с англ. – М.: Финансы и статистика, 1983.
13. Учебно-методический комплекс по курсу «Теория экономических информационных систем» / Сост. С.П. Салмин. – Н. Новгород: НКИ, 2001.
14. Цикритзис Д., Лоховский Ф. Модели данных. – М.: Финансы и статистика, 1985.

Предисловие	3
Г л а в а 1. Основные компоненты экономической информации	5
1.1. Элементарные свойства информации	5
Методические указания	5
Задания	10
1.2. Структура экономических документов	17
Методические указания	17
Задания	25
1.3. Экономические показатели	31
Методические указания	31
Задания	40
Г л а в а 2. Информационная алгебра	53
2.1. Операции над отношениями	53
Методические указания	53
Задания	60
2.2. Функциональные зависимости и ключи	66
Методические указания	66
Задания	73
2.3. Нормальные формы отношений	79
Методические указания	79
Задания	83
2.4. Ациклические базы данных	93
Методические указания	93
Задания	97
2.5. Сетевая и иерархическая модели данных	100
Методические указания	100
Задания	114
2.6. Моделирование предметных областей	122
Методические указания	122
Задания	125

Глава 3. Алгоритмы и организация данных	130
3.1. Последовательная организация данных	130
Методические указания	130
Задания	139
3.2. Цепная организация данных	143
Методические указания	143
Задания	147
3.3. Древовидная организация данных.	153
Методические указания	153
Задания	156
Глава 4. Комплексные задания и тесты	160
4.1. Комплексные задания	160
Методические указания	160
Задания	161
4.2. Список контрольных вопросов по теории экономических информационных систем	172
4.3. Тесты	176
Рекомендуемая литература	189

Учебное издание

**Мишенин Александр Иванович
Салмин Сергей Павлович**

**ТЕОРИЯ ЭКОНОМИЧЕСКИХ
ИНФОРМАЦИОННЫХ СИСТЕМ.
ПРАКТИКУМ**

Заведующая редакцией *Л.А. Табакова*
Ведущий редактор *Н.А. Кузнецова*
Младший редактор *Н.А. Федорова*
Художественный редактор *Ю.И. Артюхов*
Технический редактор *В.Ю. Фотиева*
Корректоры *Н.Н. Зубенко, Н.Б. Вторушина*
Компьютерная верстка *О.В. Фортунатовой*
Оформление художника *Н.М. Биксентеева*

ИБ № 4671

Подписано в печать 25.11.2004.
Формат 60х88 1/16. Гарнитура «Таймс». Печать офсетная
Усл. п.л. 11,76. Уч.-изд. л. 10,78. Тираж 3000 экз.
Заказ № 4126. «С»242

Издательство «Финансы и статистика»
101000, Москва, ул. Покровка, 7
Телефон (095) 925-35-02. Факс (095) 925-09-57
E-mail: mail@finstat.ru <http://www.finstat.ru>

ГП Псковской области «Великолукская городская типография»
Комитета по средствам массовой информации
182100, Великие Луки, ул. Полиграфистов, 78/12
Тел./факс: (811-53) 3-62-95
E-mail: VTL@MART.RU