

С. Ю. АЛЕКСЕЕВ, И. Л. КОРОБОВА

**РАЗРАБОТКА ИНФОРМАЦИОННОГО
ОБЕСПЕЧЕНИЯ СИСТЕМ
АВТОМАТИЗИРОВАННОЙ
ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ**

**Тамбов
Издательский центр ФГБОУ ВО «ТГТУ»
2023**

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тамбовский государственный технический университет»

С. Ю. АЛЕКСЕЕВ, И. Л. КОРОБОВА

РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ СИСТЕМ АВТОМАТИЗИРОВАННОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Утверждено Ученым советом университета
в качестве учебного пособия для бакалавров 3 курса
направления подготовки 09.03.01 «Информатика
и вычислительная техника», изучающих дисциплину
«Разработка информационного обеспечения»

Учебное электронное издание



Тамбов
Издательский центр ФГБОУ ВО «ТГТУ»
2023

УДК 004.65
ББК 32.972.134
А47

Рецензенты:

Кандидат технических наук, доцент, доцент кафедры
«Математическое моделирование и информационные технологии»
Института математики, физики и информационных технологий
ФГБОУ ВО «ТГУ им. Г. Р. Державина»
Д. С. Соловьев

Кандидат технических наук, доцент, доцент кафедры
«Информационные процессы и управление» ФГБОУ ВО «ТГТУ»
А. А. Третьяков

Алексеев, С. Ю.

А47 Разработка информационного обеспечения систем автоматизированной поддержки принятия решений [Электронный ресурс] : учебное пособие / С. Ю. Алексеев, И. Л. Коробова. – Тамбов : Изд-во ФГБОУ ВО «ТГТУ», 2023. – 1 электрон. опт. диск (CD-ROM). – Системные требования : ПК не ниже класса Pentium II ; CD-ROM-дисковод ; 1,84 Mb ; RAM ; Windows 95/98/XP ; мышь. – Загл. с экрана.

ISBN 978-5-8265-2606-4

Рассматриваются основные понятия баз данных, систем управления базами данных, приводятся общие сведения по разработке информационного обеспечения автоматизированных систем.

Предназначено для бакалавров 3 курса направления подготовки 09.03.01 «Информатика и вычислительная техника», изучающих дисциплину «Разработка информационного обеспечения».

УДК 004.65
ББК 32.972.134

*Все права на размножение и распространение в любой форме остаются за разработчиком.
Незаконное копирование и использование данного продукта запрещено.*

ISBN 978-5-8265-2606-4 © Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный технический университет» (ФГБОУ ВО «ТГТУ»), 2023

ВВЕДЕНИЕ

Работа с данными является одной из важнейших составляющих любой автоматизированной системы. Поэтому дисциплина «Разработка информационного обеспечения» является одной из ключевых в подготовке бакалавров по направлению «Информатика и вычислительная техника».

Для успешной интеграции данных в систему необходимо обладать обязательными компетенциями в области проектирования баз данных, уметь правильно пользоваться существующими программными решениями и владеть навыками разработки новых программных средств.

В первом разделе учебного пособия приводятся основные сведения о базах данных. Второй раздел посвящен обзору функций и свойств систем управления базами данных. Третий раздел пособия рассказывает о существующих моделях данных. Отдельный раздел пособия посвящен наиболее распространенным реляционным базам данных. В пятом разделе подробно рассматриваются этапы проектирования информационной системы.

В шестом разделе учебного пособия приводятся рекомендации для выполнения курсовой работы по дисциплине «Разработка информационного обеспечения».

Несомненно, учебное пособие будет полезным для студентов бакалавриата при изучении дисциплины «Разработка информационного обеспечения», а также при подготовке выпускной квалификационной работы для бакалавров и магистрантов по направлению подготовки «Информатика и вычислительная техника».

1. БАЗЫ ДАННЫХ

1.1. ИНФОРМАЦИЯ И ДАННЫЕ

Рассмотрим понятия «*информация*» и «*данные*».

Информация – это сведения об окружающем мире, которые воспринимает человек или специальное техническое средство.

Данные представляют собой зафиксированные на определенном носителе сведения, которые можно хранить, передавать, обрабатывать.

По сути, информация – это переработанные и проанализированные данные.

Данные можно рассматривать на трех уровнях: внешнем, концептуальном и внутреннем (рис. 1). Внешний уровень – это представление пользователя о предметной области. На концептуальном уровне данные представляются в виде объектов (сущностей), их свойств и связей между ними, но не учитывают способ хранения. Внутренний уровень представления данных определяет физическое представление данных.

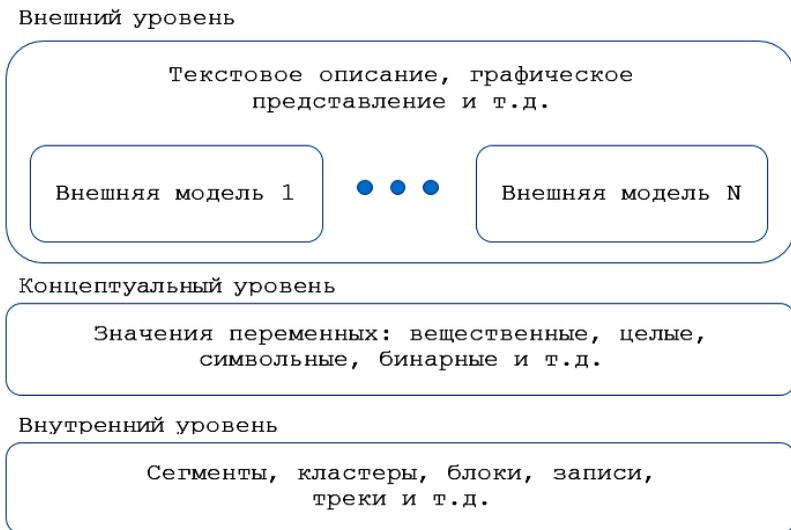


Рис. 1. Уровни представления данных

1.2. ОСНОВНЫЕ ПОНЯТИЯ

Рассмотрим основные определения понятий, часто используемых в информационном обеспечении (ИО).

База данных (БД) – структурированная совокупность данных.

Система управления базами данных (СУБД) состоит из лингвистических и программных средств, предназначенных для создания, ведения и эксплуатации баз данных.

Банк данных (БнД) – совокупность баз данных и системы управления базами данных.

Модель данных – формализованное описание, отражающее состав и типы данных, а также взаимосвязь между ними.

1.3. БАНКИ ДАННЫХ. ТРЕБОВАНИЯ. АРХИТЕКТУРА

Рассмотрим более полное определение банков данных. *Банк данных* – это система специальным образом организованных данных (баз данных), программных, технических, лингвистических, организационных, методических средств, предназначенных для обеспечения, хранения и использования данных.

Банк данных представляет собой человеко-машинную систему, включающую в свой состав различные компоненты (рис. 2).

Информационный компонент банка данных состоит из баз данных, схем баз данных, словарей данных.

Программные средства банка данных обеспечивают взаимодействие всех частей информационного обеспечения системы (рис. 3).

Лингвистические средства банков данных обеспечивают интерфейс пользователей разных категорий с данными (рис. 4).



Рис. 2. Компоненты банков данных



Рис. 3. Программные средства банков данных

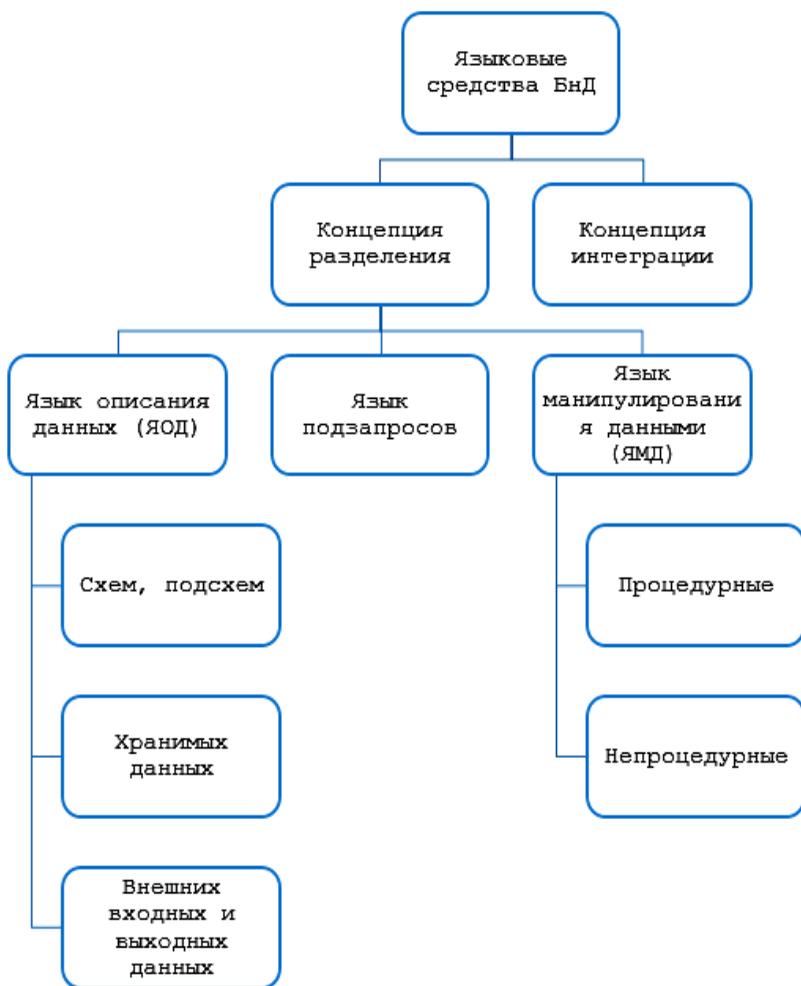


Рис. 4. Лингвистические средства банков данных

В качестве технических средств для банка данных обычно используются универсальные ЭВМ, стандартный набор периферийных устройств и сетевого оборудования.

Но возможно использование специальных технических средств, например серверы, накопители на магнитных лентах (стримеры), накопители на оптических носителях (CD R/W).

К организационно-методическим средствам БНД относятся различные инструкции, методические и регламентирующие документы для пользователей различных категорий.

Группа специалистов, обеспечивающих разработку, функционирование и развитие систем баз данных, называется администратором банка данных (АБД).

1.4. КЛАССИФИКАЦИЯ БАНКОВ ДАННЫХ

Банки данных классифицируются по различным признакам. Большинство классификационных признаков относится к базе данных. Рассмотрим некоторые из них.

По форме представления данных системы разделены на видео- и аудиосистемы, также мультимедиа, символьные. Пока наибольшее применение находят базы данных, содержащие обычные символьные данные.

По типу хранимой информации БД можно разделить на документальные, фактографические и лексикографические.

По характеру организации и хранения данных и обращению к ним различают локальные (однопользовательские), общие (интегрированные), распределенные и объектно-ориентированные.

1.5. СВОЙСТВА БАНКОВ ДАННЫХ

К основным свойствам банков данных относят:

Скорость. Время реакции, т.е. получение ответа на запрос.

Доступность. Какие данные, содержащиеся в БД, доступны данной категории пользователей.

Гибкость. Возможность получить ответ на сложные запросы.

Целостность. Снижение избыточности данных, согласованность данных при упорядочении обновления.

2. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

2.1. ФУНКЦИИ СУБД

Система управления базой данных представляет собой программное обеспечение, которое управляет доступом к данным. Это происходит следующим образом:

1. Пользователь выдает запрос на доступ, применяя определенный подязык данных.
2. СУБД получает запрос и анализирует его.
3. СУБД определяет структуру хранения для конкретного пользователя.
4. СУБД выполняет необходимые операции над храняемыми данными.

Описанные действия выполняются благодаря функциям СУБД. Рассмотрим их более подробно.

- Определение данных. СУБД определяет данные в исходной форме и преобразует в форму, пригодную для хранения.
- Обработка данных. СУБД должна уметь обрабатывать запросы пользователя для выбора, изменения или

удаления данных в базе данных или добавление новых данных.

– Безопасность и целостность данных. Достигается контролем пользовательских запросов и пресечением попыток нарушения правил безопасности и целостности данных.

– Восстановление и дублирование данных. Данная функция может быть встроена в СУБД или выполняться специальным программным компонентом.

– Словарь данных. СУБД должна обеспечивать функцию словаря данных. По сути, словарь данных – это база данных, которая содержит данные о данных, например исходные и объектные схемы внешнего и концептуального уровня, отчеты для различных пользователей и проч.

– Производительность. Все перечисленные функции должны выполняться с максимально возможной эффективностью.

2.2. СТРУКТУРА СУБД

Организация типичной СУБД и состав ее компонентов соответствуют рассмотренному нами набору функций.

Логически, например, в реляционной СУБД можно выделить наиболее внутреннюю часть – ядро СУБД, компилятор языка БД, подсистему поддержки времени выполнения и набор утилит.

3. МОДЕЛИ ДАННЫХ

СУБД организует и структурирует данные таким образом, чтобы пользователи и прикладные программы могли их сохранять и выбирать из базы данных. Структуры данных и способы доступа к ним, обеспечиваемые конкретной СУБД, называются ее моделью данных. Традиционно СУБД делятся по типу модели данных на *иерархические*, *сетевые* и *реляционные*. Такое деление моделей и СУБД основывается на характере связей между записями.

Иерархическая структура включает простые и составные компоненты, например векторы, повторяющиеся и допускает многоуровневость (рис. 5).

На рисунке 5 цифрами отмечены основные типы элементов:

- 1 – простой элемент;
- 2 – набор однотипных элементов;
- 3 – повторяющаяся группа;
- 4 – неповторяющаяся группа.

Состав записей в структуре может быть постоянным или переменным. Основными характеристиками записи являются ее тип (символьный, числовой, дата, логический и т.д.) и длина (фиксированная, переменная и неопределенная).

Рассмотрим более подробно модели данных.

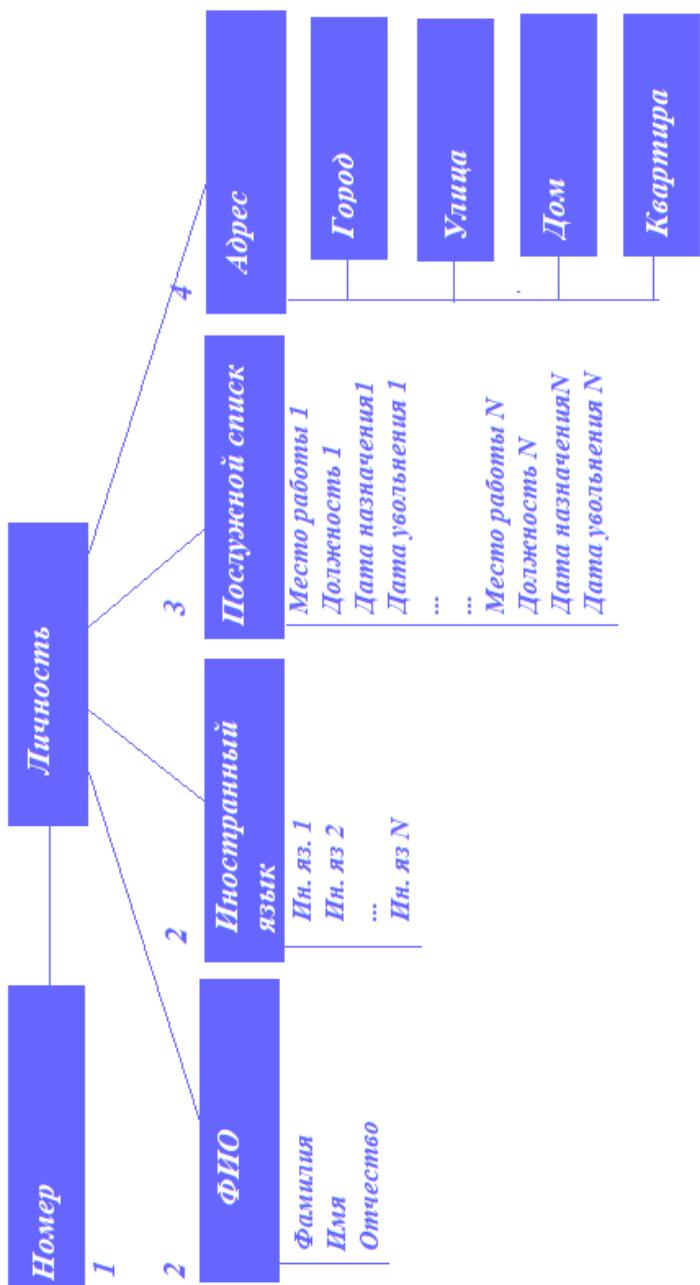


Рис. 5. Иерархическая структура записи

3.1. ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

В классических *иерархических моделях* имеется один файл, который является входом в структуру (корень дерева). Остальные файлы связаны друг с другом таким образом, что каждый из них за исключением корневой вершины имеет ровно одну исходную вершину («предок») и любое число подчиненных вершин («потомков»). Между записью файла-«предка» и записями порожденного файла имеется отношение «один ко многим» (1:M). Как частный случай может быть отношение «один к одному» (1:1). Различают также тип связи «многие ко многим» (M:M).

Для чтения информации из иерархической базы данных требуется возможность перемещения по записям, за один шаг переходя на одну запись вверх, вниз или в сторону.

Пример схемы иерархической БД «Деталь» показан на рис. 6.

Состав информации базы данных «Деталь»:

1. Для каждой детали: шифр детали (уникальный), название и краткое описание назначения детали, информация о технических характеристиках и наличии на складе.

2. Характеристики детали включают необходимые технические данные, в примере это вес детали, материал и ее габаритные размеры.

3. Получение деталей на склад характеризуется датой получения, количеством деталей и уникальным номером накладной.

4. Для каждой детали существует предприятие-изготовитель, имеющее почтовый адрес, название и шифр.

5. Деталь может быть куплена через посредников, поэтому отдельно записывается название поставщика, его почтовый адрес, цена за одну деталь и шифр поставщика.

Деталь – корневой сегмент. Остальные – подчиненные типы сегментов.

Пусть имеется один экземпляр корневого типа сегмента Деталь, один экземпляр Характеристики, два экземпляра Приход, один экземпляр Изготовитель и два экземпляра Поставщик (рис. 7). Первому экземпляру Приход подчинен один экземпляр Изготовитель и два экземпляра Поставщик. Для второго экземпляра Приход пока неизвестны Изготовитель и Поставщик, например собственное экспериментальное производство.

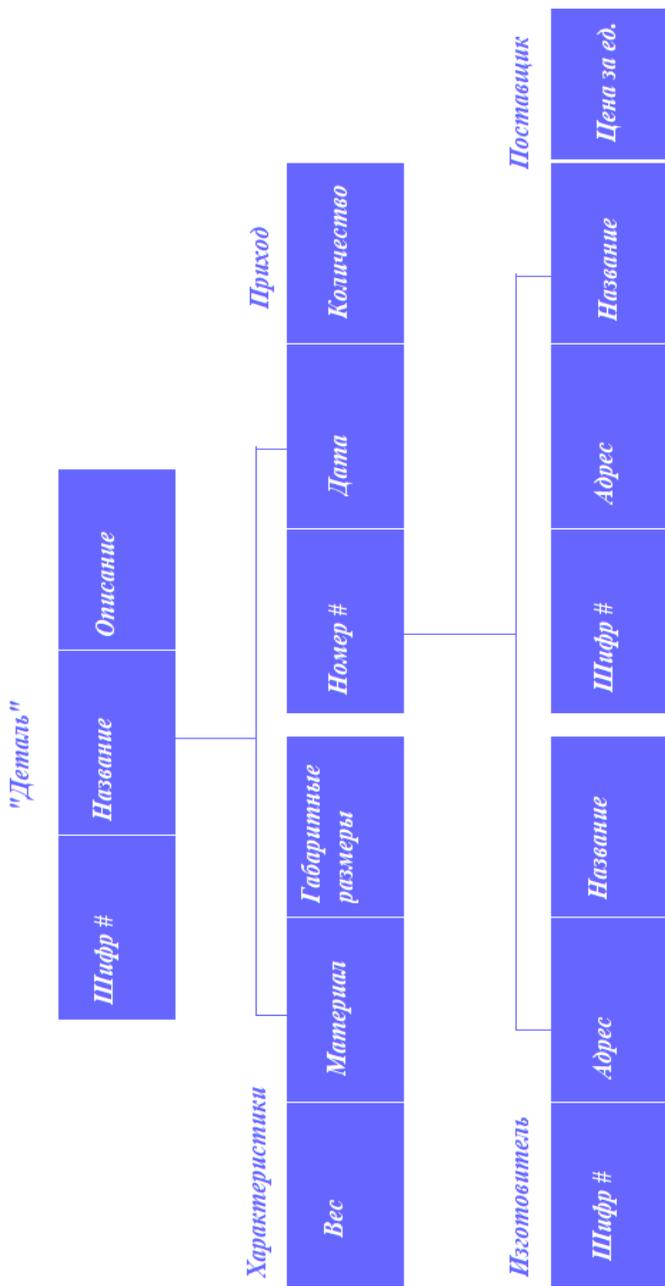


Рис. 6. Один экземпляр дерева иерархической базы данных

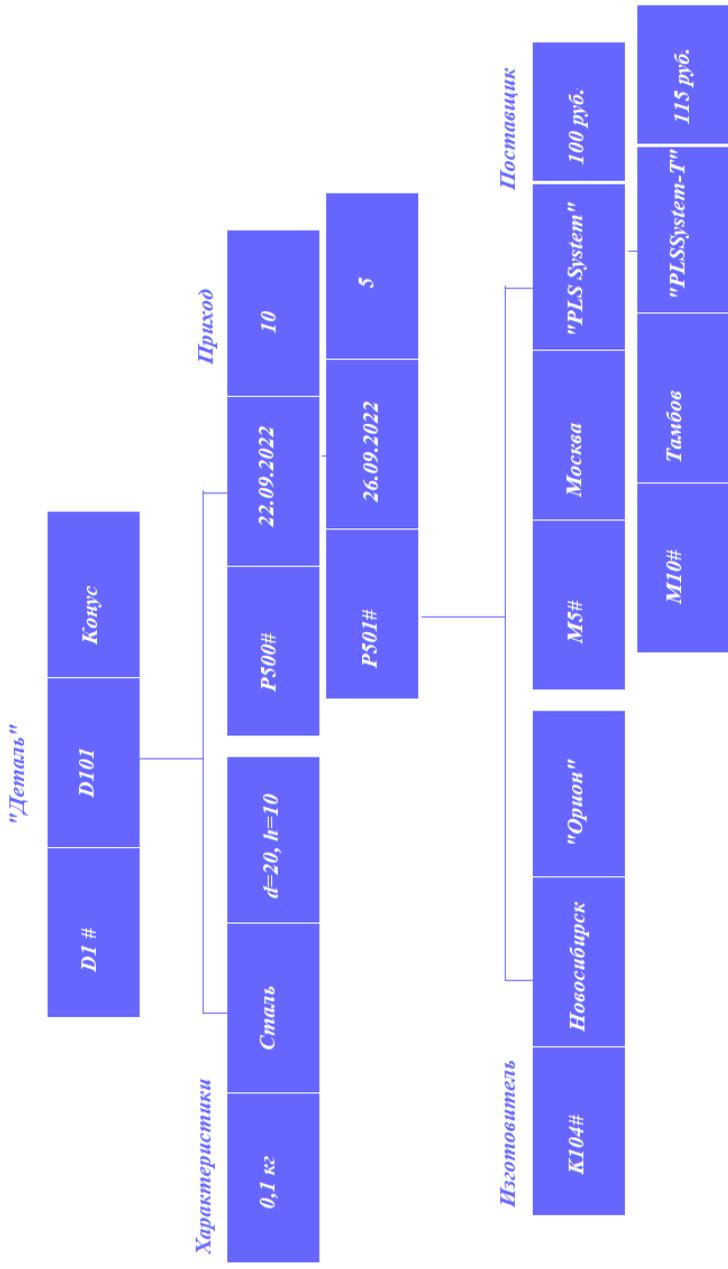


Рис. 7. Экземпляр дерева иерархической базы данных «Деталь»

Ссылки между записями различных деревьев не поддерживаются.

Иерархический порядок в БД считается очень важным, так как определяет доступ к информации. Поиск в БД осуществляется по составному ключу, т.е. для выполнения запроса надо указывать значение ключа на каждом уровне иерархии.

Например, для экземпляра сегмента Поставщик, соответствующего адресу Тамбов, значение ключа будет иметь вид: 1 D1 1 P500 2 M10.

3.2. СЕТЕВАЯ МОДЕЛЬ

В сетевых моделях, если на нее не накладывается никаких ограничений, в принципе любой файл может быть точкой входа в систему, каждый из файлов может быть связан с произвольными числами других файлов, и между записями связанных файлов могут быть отношения 1:1, 1:M, M:M. Однако в реальных СУБД на модель накладываются различные ограничения.

Во многих сетевых СУБД не поддерживается отношение M:M. В таких моделях каждая связь между парой файлов определяется отдельно, и для каждой из них

один файл в этой паре объявляется «владельцем», а другой – «членом». Отношение между записями 1:M.

Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре потомок может иметь любое число предков.

Сетевая БД состоит из набора записей и набора связей между ними. Тип связи определяется для двух типов записи: предка и потомка (рис. 8).

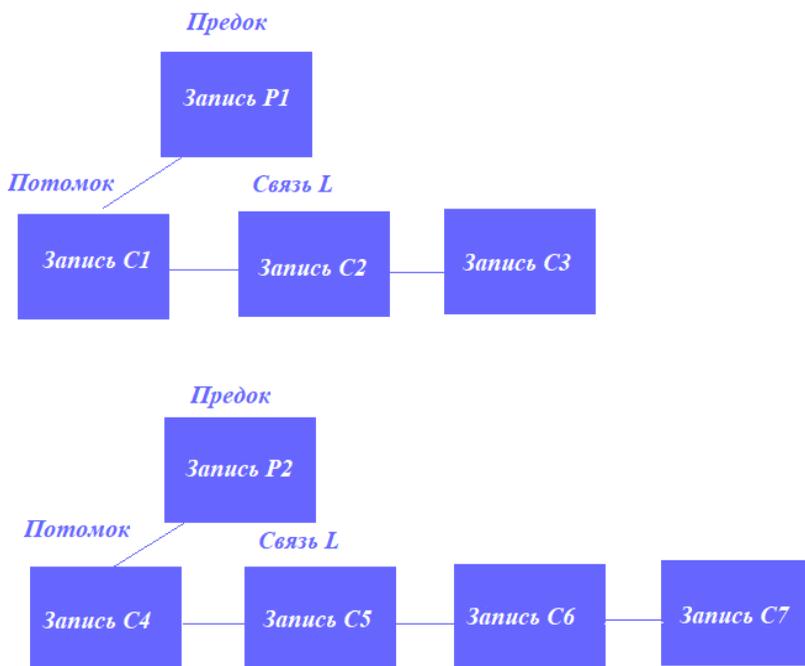


Рис. 8. Пример экземпляров набора

При доступе к информации, содержащейся в базе данных, программа может выполнить следующие задачи:

- найти определенную запись предка по ключу;
- перейти к первому потомку в определенном множестве;
- перейти от одного потомка к другому в определенном множестве по горизонтали;
- перейти вверх от потомка к его предку в другом множестве.

На рисунке 9 показан простой пример сетевой схемы БД. Примерный набор операций может быть таковым:

- найти конкретную запись в наборе однотипных записей, например программиста Иванова;
- перейти от предка к первому потомку (к первому сотруднику отдела);
- перейти к следующему потомку в некоторой связи (от Иванова к Петрову);
- перейти от потомка к предку по некоторой связи (найти отдел Иванова);
- создать новую запись;
- удалить запись;
- модифицировать запись;
- включить связь;
- исключить из связи;
- переставить в другую связь и т.д.

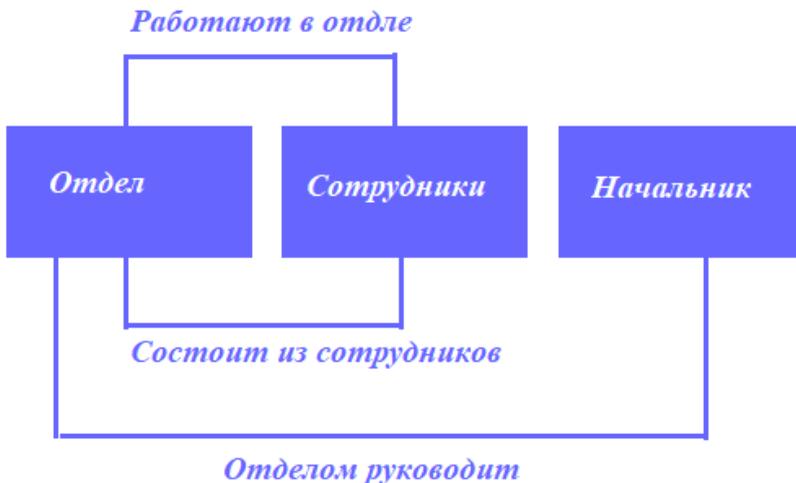


Рис. 9. Пример сетевой базы данных

3.3. ДОСТОИНСТВА И НЕДОСТАТКИ ИЕРАРХИЧЕСКИХ И СЕТЕВЫХ СУБД

Иерархические и сетевые модели имеют достоинства:

- наглядная структура;
- использование отношений «предок–потомок»;
- производительность – отношения «предок–потомок»

реализованы в виде физических указателей из одной записи на другую, вследствие чего перемещение по базе данных выполняется очень быстро.

Конечно, у сетевых и иерархических баз данных есть недостатки:

- наборы отношений и структура записей должны были быть заданы заранее и жестко;
- изменение структуры обычно требует полной перестройки базы данных;
- слишком сложно пользоваться.

3.4. РЕЛЯЦИОННАЯ МОДЕЛЬ

Реляционная модель данных, предложенная Э. Коддом, была попыткой упростить структуру базы данных. В ней отсутствовала явная структура «предок–потомок», а все данные были представлены в виде простых таблиц, разбитых на строки и столбцы. На рисунке 10 показана реляционная модель данных.

Реляционной называется БД, в которой все данные, доступные пользователю, организованы в виде таблиц, а все операции базы данных выполняются над этими таблицами.

Реляционная СУБД способна реализовать отношения «предок–потомок», однако эти отношения представлены исключительно значениями, содержащимися в таблицах базы данных (рис. 10).

PRODUCTS		
ID ТОВАРА	ОПИСАНИЕ	...
55738	Widget Remover	
65763	Reducer	
43071	Size 4 Widget	
...
26264	Size 1 Widget	

SALES REPS		
ID СОТРУДНИКА	ФИО	...
101	Серых Святослав	
301	Cooper John	
...
401	Beck Heike	

Рис. 10. Пример реляционной модели данных

Более подробно реляционная база данных описана в разделе 4 данного учебного пособия.

3.5. МНОГОМЕРНАЯ МОДЕЛЬ

Многомерная модель предназначена для аналитической обработки информации.

Многомерная модель отражает несколько измерений данных. Например, на рис. 11 приведены реляционное (*a*) и многомерное (*b*) представления одних и тех же данных.

Таблицы содержат информацию о среднем балле студентов в определенные даты. Основные понятия в многомерной модели – измерение и ячейка.

Фамилия	Дата	Средний балл
Петров	11.10	4,34
Петров	11.11	4,67
Петров	11.12	4,78
Фадеев	11.10	4,7
Фадеев	11.11	4,8
Фадеев	11.12	5,0
Иванов	11.10	3,15
Иванов	11.11	3,67
Иванов	11.12	4,15

а)

Фамилия	11.10	11.11	11.12
Петров	4,34	4,67	4,78
Фадеев	4,7	4,8	5,0
Иванов	3,15	3,67	4,15

б)

Рис. 11. Реляционное (а) и многомерное (б) представления данных

В качестве измерений многомерной модели часто используются даты (день, месяц, год и пр.), а в ячейке многомерной модели – это фиксированное значение.

Пример трехмерной модели данных приведен на рис. 12. В данном примере с помощью многомерной модели представлена информация о среднем балле студентов по определенному предмету на определенную дату. Измерения модели:

- Фамилии студентов;
- Название предмета;
- Дата.



Рис. 12. Пример трехмерной модели данных

Если используется модель с большей, чем три, размерностью, то данные представляются на срезе многомерного куба в виде двумерных таблиц. Срез получают фиксацией определенных измерений.

Наибольшую ценность многомерные модели приобретают при анализе больших данных, а также при реализации сложных запросов.

Однако при использовании в стандартных задачах хранения и обработки данных многомерная модель может оказаться сложной, громоздкой и уязвимой.

3.6. ПОСТРЕЛЯЦИОННАЯ МОДЕЛЬ

Постреляционная модель данных является расширением реляционной модели, в которой допускаются неоднозначные поля. Набор значений в таких полях считается дополнительной таблицей, встроенной в основную.

Накладные

Номер_накладной	Номер_покупателя
0373	8723
8374	8232
7364	8723

Накладные товары

Номер_накладной	Название_товара	Количество_товара
0373	Сыр	3
0373	Рыба	2
8374	Лимонад	1
8374	Сок	6
8374	Печенье	2
7364	Йогурт	1

а)

Накладные

Номер_накладной	Номер_покупателя	Название_товара	Количество_товара
0373	8723	Сыр	3
		Рыба	2
8374	8232	Лимонад	1
		Сок	6
		Печенье	2
7364	8723	Йогурт	1

б)

Рис. 13. Пример представления информации с помощью реляционной (а) и постреляционной (б) моделей

Постреляционной таблицей можно связать несколько реляционных таблиц. Но следует помнить, что при использовании постреляционной модели данных требуется принимать дополнительные меры для обеспечения их целостности.

На рисунке 13 приведен пример представления одной и той же информации с помощью реляционной (*a*) и постреляционной (*б*) моделей. В первом случае данные хранятся в рамках двух таблиц, а во втором – в одной.

Длина и количество полей не является константой, поэтому структура данных и таблиц имеют большую гибкость.

3.7. ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

Теоретическую основу ранее перечисленных моделей данных составляет теория множеств.

А вот в основе объектно-ориентированной модели данных лежат положения объектно-ориентированного программирования. В таких моделях данные представляются в виде объектов, с которым связано уникальное имя, состояние и поведение.

Состояние объекта – это его свойства, а поведение объекта – это процедуры, которые выполняют действия над атрибутами объекта в различных ситуациях. Объекты

могут объединяться в классы. При определении класс определяются методы.

Для этой модели справедливы все объектно-ориентированные положения: наследование, инкапсуляция, полиморфизм.

Например, для примера базы данных на рис. 14 «БИБЛИОТЕКА» – родитель для «АБОНЕМЕНТ», «КАТАЛОГ», «ВЫДАЧА»; «КАТАЛОГ» – родитель для «КНИГА». Все свойства и методы от родителя распространяются на всех потомков. Так проявляется наследование.

Инкапсуляция означает, что состояние определяется тем объектом, в который оно инкапсулировано.

Полиморфизм допускает в объектах разных типов иметь одни методы. Например, для базы данных на рис. 14 объекты класса «КНИГА», имеющие разных родителей из класса «КАТАЛОГ», могут иметь разный набор свойств.

С помощью объектно-ориентированной модели можно отображать сложные взаимосвязи между объектами. В них можно хранить не только структуру данных, но их поведение. Однако, большие данные с помощью такой модели обрабатывать неудобно. Кроме того, выполнение запросов менее эффективно, а отсутствие теоретической базы усложняет понимание сути объектно-ориентированной модели данных.

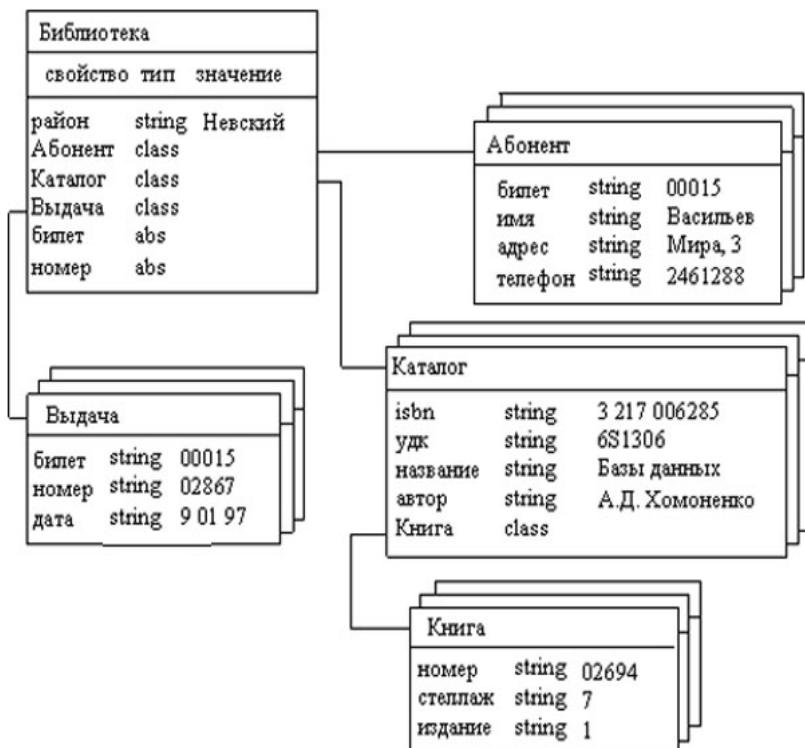


Рис. 14. Пример объектно-ориентированной модели данных

4. РЕЛЯЦИОННАЯ БАЗА ДАННЫХ

Рассмотрим пример реляционной базы данных для приложения, выполняющего обработку заказов (рис. 15).

В базе данных содержится пять таблиц, которые хранят информацию о товарах, организациях, работниках, клиентах и заказах.

4.1. ТАБЛИЦЫ

В реляционной БД информация организована в виде таблиц, разделенных на строки и столбцы, в ячейках которых содержатся значения данных. Каждая таблица имеет уникальное имя, описывающее ее содержимое. Более наглядно структуру таблицы иллюстрирует рис. 16, на котором изображена таблица OFFICES. Каждая физическая сущность – один офис. Пять строк таблицы – пять офисов компании. Каждый вертикальный столбец – один элемент данных для каждого из офисов. Например, в столбце ГОРОД содержатся названия городов, в которых расположены офисы. На пересечении строки и столбца таблицы содержится только одно значение данных. Это могут быть строки, числа, идентификаторы и т.д.

PRODUCTS					
ID_ТОВАРА	ОПИСАНИЕ	ЦЕНА_ЕД	НАЛИЧИЕ	ПРОИЗВОДИТЕЛЬ	
55738	Widget Remover	53	106	США	
65763	Reducer	28	325	Россия	
43071	Size 4 Widget	78	9	США	

SALES_REPS					
ID_СОТРУДНИКА	ФИО	ВОЗРАСТ	ID_ОФИСА		
101	Серых Святослав	36	100		
301	Cooper John	25	300		
201	Bertrand Raymonde	38	200		

OFFICES					
ID_ОфИСА	ГОРОД	СТРАНА	ПРОДАЖИ	ID_СОТРУДНИКА	
100	Москва	Россия	222800	101	
200	Париж	Франция	199089	201	
300	Лондон	Великобритания	195383	301	

CUSTOMERS			
ID_КЛИЕНТА	ФИО	ID_СОТРУДНИКА	
R1516	Сухих Ростислав	101	
F2935	Primeau Josiane	201	
A3579	Mathews Richard	301	

ORDERS						
ID_ЗАКАЗА	ДАТА	ID_ТОВАРА	КОЛ_ВО	СТОИМОСТЬ	ID_КЛИЕНТА	ID_СОТРУДНИКА
#3e6764	23.10.2017	55738	3	159	R1516	101
#12197e	23.09.2017	65763	14	392	F2935	201
#80b1ae	07.12.2017	43071	25	1950	A3579	301

Рис. 15. Пример базы данных

Множество значений, которые могут содержаться в столбце, называется доменом этого столбца. Доменом столбца ГОРОД является множество всех названий городов.

Столбцы в одной таблице должны иметь уникальные имена, однако разрешается присваивать одинаковые имена столбцам, расположенным в различных таблицах.

Порядок столбцов определяется при создании таблицы. Строки таблицы в общем случае не упорядочены.

Допускается пустая таблица (количество строк равно нулю).

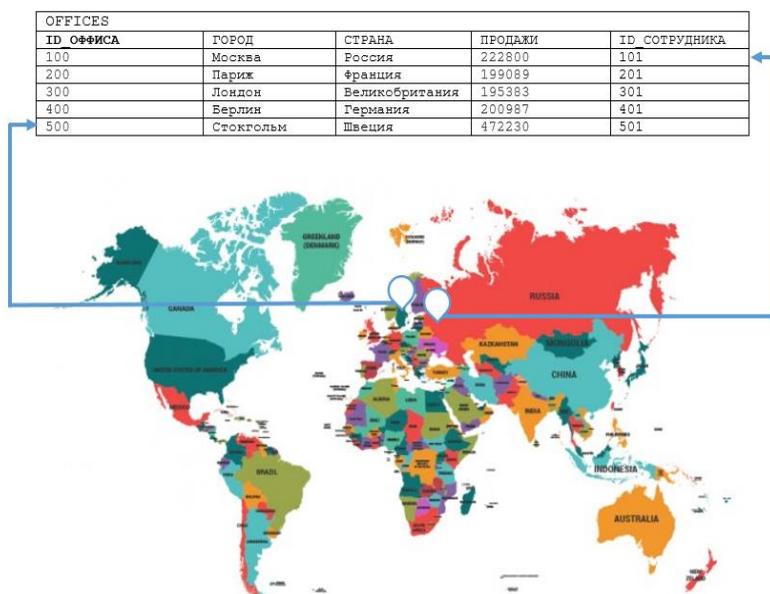


Рис. 16. Структура реляционной таблицы

4.2. ПЕРВИЧНЫЕ КЛЮЧИ

Так как строки в таблице не упорядочены, нельзя выбрать строку по ее номеру в таблице. Для выбора нужной информации используется первичный ключ (PRIMARY KEY), т.е. столбец (или несколько столбцов), для которого значения во всех строках различны.

Посмотрим на базу данных, изображенную на рис. 15. На практике в качестве первичных ключей таблиц обычно выбирают идентификаторы, такие как идентификатор офиса, сотрудника, клиента, заказа.

4.3. ВНЕШНИЕ КЛЮЧИ

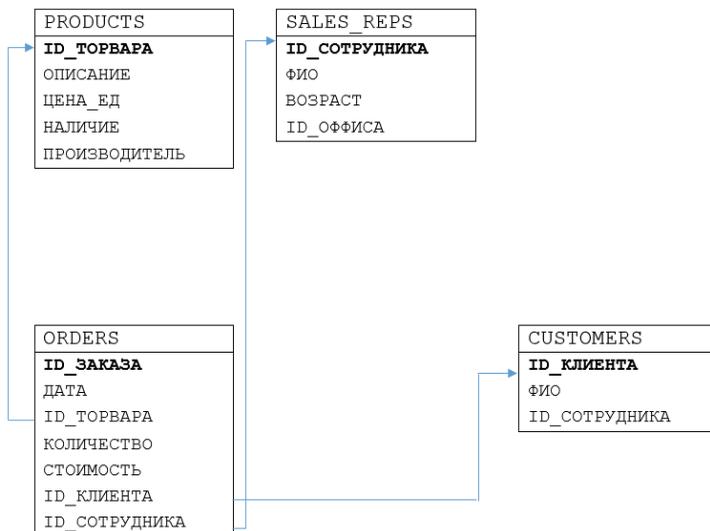
Столбец одной таблицы, значения в котором совпадают со значениями столбца, являющегося первичным ключом другой таблицы, называется внешним ключом (foreign key).

На рисунке 17 столбец ID_ОФФИСА таблицы SALES_REPS является внешним ключом для таблицы OFFICES. Хотя столбец ID_ОФФИСА и находится в таблице SALES_REPS, значения, содержащиеся в нем, представляют собой идентификаторы офисов. Эти значения соответствуют значениям в столбце ID_ОФФИСА, который является первичным ключом таблицы OFFICES. Первичный и внешний ключи создают между таблицами,

в которых они содержатся, такое же отношение «предок–потомок». Внешний ключ, как и первичный, тоже может представлять собой комбинацию столбцов. Если таблица связана с несколькими другими таблицами, она может иметь несколько внешних ключей (рис. 18).



**Рис. 17. Отношение «предок–потомок»
в реляционной базе данных**



**Рис. 18. Множественные отношения «предок–потомок»
в реляционной базе данных**

5. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

5.1. ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

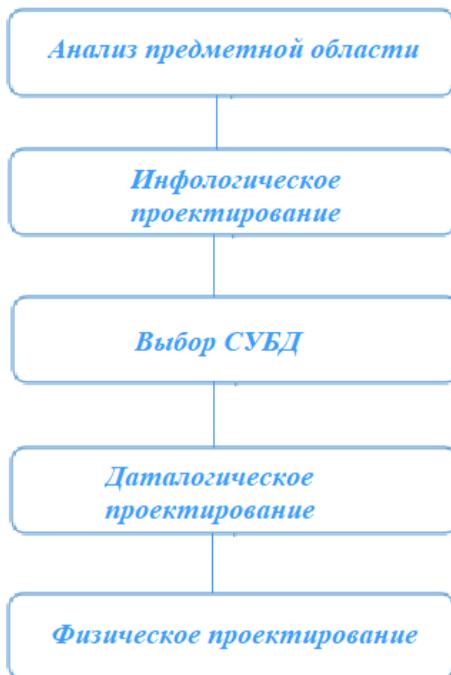


Рис. 19. Этапы проектирования

Процесс проектирования базы данных включает пять основных этапов проектирования (рис. 19):

1. Анализ предметной области. На данном этапе необходимо провести подробное словесное описание объ-

ектов предметной области и реальных связей, присутствующих между реальными объектами. Желательно определить взаимосвязи между объектами предметной области. Результатом анализа будет подробное описание информации об объектах предметной области, которая должна храниться в базе данных, формулировкой конкретных задач, которые будут решаться с использованием конкретной базы данных с кратким описанием алгоритмов их решения, описанием выходных и входных документов при работе с базы данных.

2. Инфологическое проектирование. Данный этап состоит в описании и построении схем отражений предметной области, выполненном без ориентации на используемые в дальнейшем программные и технические средства. На сегодняшний день наиболее широкое распространение получила модель П. Чена «Сущность–связь» (Entity Relationship), так называемая ER-модель (рис. 20).



Рис. 20. Пример инфологической модели

3. Выбор системы управления базой данных. Выбор СУБД зависит от опыта разработчиков, осуществляется на основе требований к базе данных и, конечно, возможностей СУБД.

4. Даталогическое проектирование. Даталогическое проектирование есть описание базы данных в терминах принятой логической модели данных. В реляционных БД даталогическое проектирование приводит к разработке схемы базы данных, т.е. совокупности схем отношений, которые адекватно моделируют объекты предметной области и семантические связи между объектами (рис. 21). Основой анализа корректности схемы являются функциональные зависимости между атрибутами базы данных. В некоторых случаях между атрибутами отношений могут появиться нежелательные зависимости, которые вызывают нежелательные последствия при модификации базы данных. Под модификацией понимают внесение новых данных, удаление данных, а также обновление значений некоторых атрибутов. Для ликвидации возможных последствий предполагается проведение нормализации отношений базы данных.

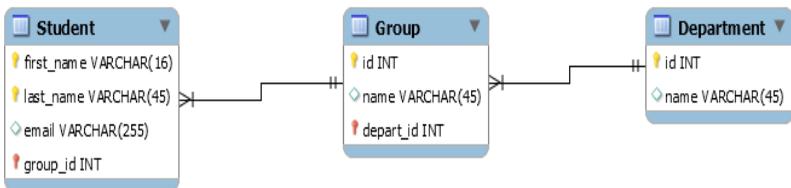


Рис. 21. Пример даталогической модели данных

5. Физическое проектирование. Этап физического проектирования заключается в определении схемы хранения, т.е. физической структуры БД. Схема хранения зависит от той физической структуры, которую поддерживает выбранная СУБД. Физическая структура должна адекватно отражать логическую структуру базы данных, а также должна обеспечивать эффективное размещение данных и быстрый доступ к ним. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL, Data Definition Language) выбранной СУБД. Принятые на этом этапе решения оказывают огромное влияние на производительность системы. Одной из важнейших составляющих проекта базы данных является разработка средств защиты. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев на этапе физического проектирования разрабатывается стратегия резервного копирования

ния. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа, набор которых также является составной частью проекта базы данных.

5.2. ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Инфологическое (концептуальное) проектирование – процесс создания модели данных о предметной области, не зависящий от любых физических аспектов ее представления.

В данном случае используется информация, полученная от пользователей. Инфологическое проектирование базы данных не зависит от выбранной СУБД, создаваемых прикладных программ, языков программирования, технических средств и т.п. При разработке инфологическая модель подвергается проверке на соответствие требованиям пользователей, и, если необходимо, модифицируется. От качества разработанной инфологической модели во многом зависит эффективность базы данных.

Несомненно, важным компонентом инфологического проектирования является модель «сущность–отношение» или «ER (Entity Relationship)-модель». На основе семанти-

ческой информации она описывает объекты предметной области и связи между ними. Затем из этого описания можно создавать любые модели данных (сетевые, объектные, реляционные, иерархические). Таким образом, ее саму можно считать общей моделью.

ER-модель предложена Питером Ченом в 1976 году. Ее свойством является наглядность представления в виде графической схемы, что является плюсом использования для анализа предметной области.

ER-диаграммы используют графическое изображение (рис. 22):

- сущностей предметной области в виде прямоугольников;
- свойств (атрибутов) – в виде овалов;
- взаимосвязей между сущностями – в виде ромбов.

Создание инфологической модели начинают с определения сущностей. Сущность представляет собой множество объектов реального мира с одинаковым набором атрибутов. На рисунке 22 определены четыре сущности: Пользователи, Клиенты, Персональные данные и Адреса. Каждая сущность идентифицируется именем и набором атрибутов (свойств). Значения атрибутов являются основным содержанием базы данных. Например, на рис. 22 для сущности Клиенты определены шесть атрибутов: Фамилия, Имя, Отчество, Пол, Телефон и id_client.

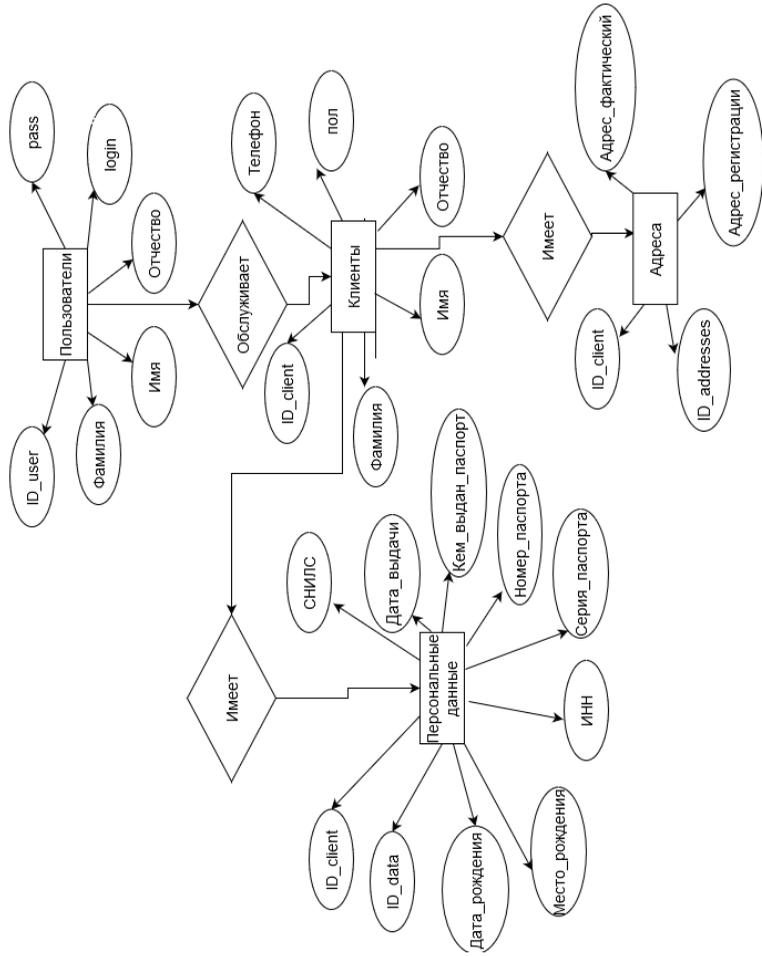


Рис. 22. Пример ER-модели

Атрибуты классифицируют следующим образом:

1. Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты определяют уникальные значения для описания конкретных сущностей.

2. Составные и простые атрибуты. Простой атрибут описывается одним, неделимым значением компонента; составной атрибут имеет множество значений компонентов, которые могут принадлежать разным типам данных.

3. Однозначные и многозначные атрибуты могут иметь как одно, так и несколько значений для каждого экземпляра сущности соответственно.

4. Основные и производные атрибуты. Значение основного атрибута является независимым от других. Значение производного атрибута получают, учитывая значения других.

5. Обязательные и необязательные. Для обязательного – значение атрибута должно быть указано при занесении в базу данных, а для необязательных допускается присвоения NULL-значений.

Для однозначной идентификации сущности выделяется набор атрибутов, называемый ключом. В наборе атрибутов Сущности можно выделить несколько потенциальных ключей. Потенциальный ключ, используемый реально для идентификации сущности, называется первичным ключом.

На рисунке 22 в качестве первичного ключа для сущности Клиент в качестве первичного ключа выбран атрибут `id_client`.

Соединение двух или более сущностей называется связью. Связи также имеют имя. На ER-диаграммах связь изображается в виде ромба, помеченного соответствующим именем. Например, на рис. 22 сущности Пользователи и Клиенты имеют связь Обслуживает.

Степень связи – количество сущностей, которые охвачены данной связью.

Если связь определена между двумя сущностями, то ее степень – 2, а называется такая связь бинарной. В общем случае связь между n сущностями называется n -арной. Возможна рекурсивная связь – связь, в которой одни и те же сущности участвуют несколько раз в разных ролях. Иногда ее называют унарной.

Семантика предметной области определяет мощность связей между сущностями. Связи бывают следующих видов:

1. Связь 1:1 (один к одному): в каждый момент времени каждому экземпляру сущности X соответствует не более одного экземпляра сущности Y . Пример такой связи приведен на рис. 23.



Рис. 23. Пример связи 1:1

2. Связь 1:М (один ко многим): каждому экземпляру сущности X соответствует любое количество экземпляров сущности Y . Пример такой связи приведен на рис. 24.



Рис. 24. Пример связи 1:М

3. Связь М:N (многие ко многим): каждому экземпляру сущности X соответствует любое количество экземпляров сущности Y , а каждому экземпляру сущности Y соответствует любое количество экземпляров сущности X . Пример такой связи приведен на рис. 25.



Рис. 25. Пример связи М:N

Описание реальных баз данных с помощью ER-модели бывает затруднительным из-за большого числа сущностей, атрибутов и связей. Поэтому ER-модель используется для описания небольших фрагментов баз данных. А для больших баз данных можно использовать язык инфологического моделирования (ЯИМ). Инфологическая модель можно представить в виде последовательности предложений для описания сущностей и связей:

Сущность (атрибут 1, атрибут 2, ..., атрибут N)

Связь [Сущность 1, S_1 , Сущность 2, S_2 , ...]

(атрибут 1, атрибут 2, ..., атрибут N),

где S_i – мощность соответствующей связи. Атрибут, входящий в ключ, отмечается подчеркиванием.

5.3. ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Конечным результатом даталогического проектирования является описание даталогической модели базы данных. В ней определяются все информационные объекты и связи между ними, типы данных и количественные характеристики. Однако не все виды связей могут отображаться в модели, например те, которые не поддерживает

конкретная СУБД. На этом этапе принимается соглашение о том, какие данные будут храниться в базе данных, а какие могут быть получены в результате выполнения запросов. На этом этапе проверяется информационная модель предметной области, исключаются одинаковые идентификаторы различных объектов, даются новые имена, определяются количество и структура файлов.

Также необходимо спроектировать структуру данных. Этот процесс может быть результатом выполнения следующих этапов:

- 1) сбор информации об объектах предметной области в одной таблице, а затем ее декомпозиция на несколько связанных таблиц на основе процедуры нормализации отношений;

- 2) формулирование знаний о системе (определение типов данных, их взаимосвязей) и рекомендаций к обработке данных, получение готовой схемы БД;

- 3) структурирование информации для использования в информационной системе в процессе проведения системного анализа.

На рисунке 26 приведен фрагмент даталогической модели базы данных, ER-модель которой приведена на рис. 22.

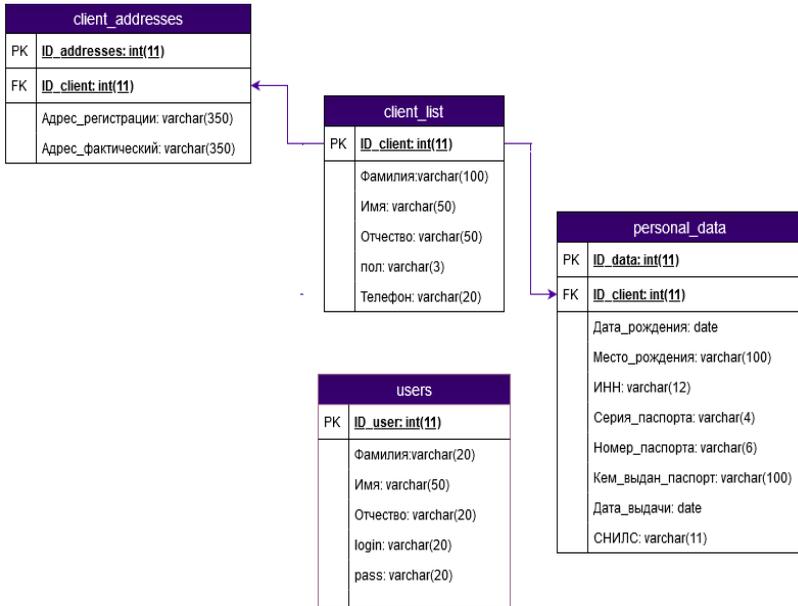


Рис. 26. Пример даталогической модели базы данных

Даталогическая модель представлена в виде набора таблиц. Каждой таблице поставлены в соответствие ее ключи. Первичный ключ (PK) для каждой таблицы должен быть только один. Вторичные ключи (FK) используются для реализации связей.

База данных включает в себя три таблицы, связанные между собой:

- client_list;
- client_addresses;
- personal_data.

Также в состав рассматриваемой базы данных входит еще одна таблица «users», не связанная с вышеперечисленными таблицами.

Таблица «users» содержит информацию о сотруднике кредитной организации («Фамилия», «Имя», «Отчество», «login» и «pass»). Столбцы «login» и «pass» содержат идентификатор пользователя и пароль, необходимые для авторизации и дальнейшего взаимодействия с приложением для работы с базой данных.

Таблица «client_list» является родительской таблицей, содержащей основную информацию о клиенте. Первичным ключом данной таблицы является столбец ID_client, имеющий атрибут AUTO_INCREMENT для генерации уникального значения идентификатора. Информация, хранящаяся в этой таблице, отражает фамилию, имя и отчество заемщика, а также пол и номер телефона. Дочерними таблицами являются следующие: «client_addresses», «personal_data», «data_work», «other_information».

Данные об адресах клиента (адрес регистрации и фактический адрес) отражены в таблице «client_addresses». Таблица «personal_data» содержит информацию о дате рождения клиента, его паспортные данные, ИНН и СНИЛС.

Для отображения даталогической модели используется нотация IDEF1X.

В рамках стандарта IDEF1X сущность изображается в виде прямоугольника, разделенного внутри на две части. В верхней части записываются ключевые атрибуты, в нижней – неключевые. Название сущности записывается над прямоугольником. Линии обозначают идентифицирующие связи.

6. ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

6.1. ДУБЛИРОВАНИЕ ДАННЫХ

В ходе проектирования реляционной базы данных данной предметной области при одинаковом наборе атрибутов могут быть выбраны различные варианты даталогической схемы.

Одной из проблем является дублирование данных. Дублирования бывают избыточными и неизбыточными, и избыточное дублирование может привести к серьезным проблемам при работе с данными. Могут возникнуть аномалии дублирования, которые приводят к усложнению работы с базой данных. Существуют аномалии модификации (возникают, когда изменения одного элемента таблицы могут привести к просмотру всей таблицы и изменению других элементов); аномалии удаления (возникают, когда при удалении одного элемента данных могут пропасть другие данные, не связанные напрямую); аномалии добавления (возникают, когда невозможно поместить новые данные, а возможно их добавление потребует просмотра всей таблицы).

Рассмотрим простой пример.

На рисунке 27 приведено отношение СОТРУДНИК (Фамилия, Отдел, Начальник отдела). Явно видно, что у каждого отдела один начальник и запись его в каждой строке приводит к избыточному дублированию.

Чтобы избавиться от дублирования, необходимо разделить это отношение на два: Сотрудник (Фамилия, Отдел) и Отдел (Отдел, Начальник отдела) (рис. 28). Такой прием называется декомпозицией.

Сотрудник

Фамилия	Отдел	Начальник отдела
Иванов	А1	Фёдоров
Петров	А1	Фёдоров
Сидоров	Б2	Блинов
Миронов	В3	Авдеев
Сергеев	Б2	Блинов

Рис. 27. Пример избыточных данных

Сотрудник		Отдел	
Фамилия	Отдел	Отдел	Начальник отдела
Иванов	А1	А1	Фёдоров
Петров	А1	В3	Авдеев
Сидоров	Б2	Б2	Блинов
Миронов	В3		
Сергеев	Б2		

Рис. 28. Пример декомпозиции

6.2. НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ

Нормальная форма – требование, предъявляемое к структуре таблиц для устранения из базы избыточных функциональных зависимостей между атрибутами.

Метод нормальных форм состоит в сборе информации о объектах решения задачи для одного отношения и последующей декомпозиции этого отношения на несколько взаимосвязанных отношений на основе правил нормализации.

Процесс проектирования базы данных с использованием нормальных форм заключается в последовательном переводе отношения из первой нормальной формы в нормальную форму более высокого порядка по определенным правилам.

Для первой нормальной формы (1НФ) характерны простые атрибуты, скалярные значения атрибутов, отсутствие повторений строк.

Например, на рис. 29 приведена таблица «Автомобили».

Очевидно нарушение нормализации 1НФ для строки BMW, так как в одной ячейке содержится список из трех элементов: M5, X5M, M1, т.е. он не является скалярным. Преобразуем таблицу к 1НФ (рис. 30).

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Рис. 29. Пример таблицы

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

Рис. 30. Пример 1НФ

Рассмотрим еще один пример. На рисунке 31 приведена таблица в 1НФ.

<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%

Рис. 31. Пример таблицы в 1НФ

Первичным ключом данной таблицы является Модель и Фирма. Однако при анализе выясняется, что цена зависит от первичного ключа, но скидка зависит только от Фирмы. Тогда приводим данную таблицу ко второй нормальной форме (2НФ) (рис. 32). То есть проводим декомпозицию на два отношения, в которых не ключевые атрибуты зависят от первичного ключа.

<u>Фирма</u>	Скидка
BMW	5%
Nissan	10%

<u>Модель</u>	<u>Фирма</u>	Цена
M5	BMW	5500000
X5M	BMW	6000000
M1	BMW	2500000
GT-R	Nissan	5000000

Рис. 32. Пример 2НФ

Переходим к третьей нормальной форме (3НФ). Она требует, чтобы все отношения в одной таблице были транзитивно зависимы от первичного ключа. На рисунке 33 приведена таблица в 2НФ.

<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	+7 (123) 45 67 890
Audi	Риал-авто	+7 (123) 45 67 890
Nissan	Некст-Авто	+7 (945) 24 12 123

Рис. 33. Пример таблицы в 2НФ

<u>Магазин</u>	Телефон
Риал-авто	+7 (123) 45 67 890
Некст-Авто	+7 (945) 24 12 123

<u>Модель</u>	Магазин
BMW	Риал-авто
Audi	Риал-авто
Nissan	Некст-Авто

Рис. 34. Пример ЗНФ

Первичным ключом является Модель. Однако атрибут Телефон напрямую не зависит от ключа. Оно зависит только от Магазина. То есть зависимость Модель–Телефон является транзитивным. Проведем декомпозицию отношения и получим два отношения, находящихся в ЗНФ (рис. 34).

Для третьей нормальной формы существует расширение – нормальная форма Бойса–Кодда (НФБК). Необходимость в ней возникает, если отношение имеет два (или более) возможных ключа, которые являются составными и имеют общий атрибут. Например, для таблицы на рис. 35 потенциальными ключами могут быть: (Площадка, Начало занятия); (Площадка, Конец занятия); (Абонемент, Начало занятия); (Абонемент, Конец занятия). Однако есть зависимость (Площадка, Абонемент), в которой левая часть не является потенциальным ключом отношения, т.е. отношение не находится в нормальной форме Бойса–Кодда.

Площадка	Начало занятия	Конец занятия	Абонемент
Баскетбол	9:00	10:00	Баскетбольный - детский
Футбол	10:00	11:00	Футбольный-взрослый
Футбол	11:00	12:00	Футбольный - детский
Баскетбол	12:00	13:00	Баскетбольный- взрослый
Футбол	13:00	14:00	Футбольный - детский

Рис. 35. Пример таблицы в ЗНФ

Проведем декомпозицию, чтобы привести эту таблицу к НФБК (рис. 36).

<u>Абонемент</u>	<u>Начало занятия</u>	<u>Конец занятия</u>
Баскетбольный - детский	9:00	10:00
Футбольный-взрослый	10:00	11:00
Футбольный - детский	11:00	12:00
Баскетбольный- взрослый	12:00	13:00
Футбольный - детский	13:00	14:00

Рис. 36. Пример НФБК

Площадка	<u>Абонемент</u>
Баскетбол	Баскетбольный - детский
Футбол	Футбольный-взрослый
Футбол	Футбольный - детский
Баскетбол	Баскетбольный- взрослый

Рис. 36. Окончание

Перейдем к четвертой нормальной форме (4НФ). Предположим, у нас есть база данных книг в разных сетях магазинов в разных городах (рис. 37).

Сеть магазинов	Название книги	Расположение магазина
«Шестёрочка»	«Колобок»	Москва
«Книжный»	«Война и мир»	Петербург
«Книжный»	«Колобок»	Петербург
«Читайка»	«Властелин Колец»	Москва
«Читайка»	«Война и мир»	Москва
«Читайка»	«Властелин Колец»	Петербург
«Букварик»	«Война и мир»	Москва
«Букварик»	«Война и мир»	Петербург

Рис. 37. Пример таблицы в НФБК

Поскольку эта структура таблицы состоит из составного первичного ключа, она не содержит неключевых атрибутов и уже находится в НФБК. Однако, если мы предположим, что все доступные книги предлагаются в каждой области, мы можем заметить, что «Название книги» неоднозначно связано с определенным «Расположение магазина», и поэтому таблица не удовлетворяет четвертой нормальной форме, которая требует, чтобы отношение не содержало нетривиальных многозначных зависимостей.

Нормализация отношения в 4НФ представлено на рис. 38.

Существует ряд дополнительных нормальных форм, например 5НФ и 6НФ, однако вряд ли они встретятся в реальных предметных областях и в этом учебном пособии не рассматриваются.

Создание нормализованных отношений является важнейшим этапом проектирования баз данных, но стоит так же отметить, что по мере преобразования может расти и количество таблиц, что ведет к усложнению структуры базы данных и снижению скорости работы с ней.

Если количество запросов на выборку существенно превышает количество запросов на обновление, то целесообразно проанализировать базу данных на возможность денормализации для увеличения ее производительности.

<u>Сеть магазинов</u>	Название книги
«Шестёрочка»	«Колобок»
«Книжный»	«Война и мир»
«Книжный»	«Колобок»
«Читайка»	«Властелин Колец»
«Читайка»	«Война и мир»
«Букварик»	«Война и мир»

<u>Сеть магазинов</u>	Расположение магазина
«Шестёрочка»	Москва
«Книжный»	Петербург
«Читайка»	Москва
«Читайка»	Петербург
«Букварик»	Москва
«Букварик»	Петербург

Рис. 38. Пример 4НФ

Но стоит помнить, что за счет увеличения производительности увеличивается и избыточность, это ведет к усложнению создания приложений. На практике большинство баз данных нормализуют до третьей или четвертой нормальных форм.

7. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

В соответствии с учебным планом подготовки бакалавров по направлению «Информатика и вычислительная техника» изучение дисциплины «Разработка информационного обеспечения» включает выполнение курсовой работы.

В данном разделе кратко определим требования и содержание курсовой работы.

7.1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Курсовая работа представляет собой завершающий этап изучения дисциплины «Разработка информационного обеспечения». Целью работы являются обобщение и систематизация полученных знаний, получение практических навыков при разработке информационного обеспечения автоматизированной системы, разработка программных продуктов для реализации баз данных.

7.2. ОБЩИЕ ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВЫХ РАБОТ ПО ДИСЦИПЛИНЕ «РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ»

Для сдачи курсовой работы студент готовит следую-

I. Диск, на котором записано три файла в формате .doc/.docx:

– ТГТУ 09.03.01.00N КР ДЭ-ТЛ.doc (*N – номер студента по списку*) – Титульный лист.

– ТГТУ 09.03.01.00N КР ДЭ-ЗД.doc (*N – номер студента по списку*) – Задание на курсовую работу.

– ТГТУ 09.03.01.00N КР ДЭ-ПЗ.doc (*N – номер студента по списку*) – Пояснительная записка.

Пояснительная записка включает следующие разделы:

Содержание (название разделов, подразделов с указанием страниц; титульный лист не нумеруется, но считается).

Введение (отразить актуальность и общее назначение разработки, указать сроки разработки).

1. Анализ предметной области (необходимо провести анализ информации, которая должна храниться в базе данных; выявить типы пользователей: определить основные объекты предметной области и связи между ними).

2. Постановка задачи проектирования (основываясь на специфике решаемой задачи, определяются требования к содержанию и функциональности системы для различных категорий пользователей).

3. Разработка информационного обеспечения

В данном разделе следует описать этапы проектирования базы данных.

3.1. Описание даталогической модели. На основе анализа предметной области определяются основные сущности, свойства и связи. Выбирается система управления базой данных и разрабатывается даталогическая модель данных. Даталогическую модель базы данных следует вынести в Приложение.

3.2. Описание структурной схемы. Следует разделить систему на отдельные подсистемы, каждая из которых предназначена для решения определенных задач, например подсистемы ввода-вывода, визуализации, расчета параметров, поддержки принятия решений и пр. Для каждой подсистемы приводятся все виды обеспечений (программное, информационное, лингвистическое, математическое, методическое, техническое). Определяются общесистемные ресурсы. Структурная схема приводится в Приложении.

3.3. Описание функциональной схемы. Следует привести последовательность шагов для решения всех поставленных в разделе 2 задач. Функциональная схема приводится в Приложении.

4. Реализация информационного обеспечения

В этом разделе рассматриваются вопросы реализации программного продукта. Выбирается целевая платформа, на которой будет функционировать система, подбираются программные средства, необходимые для разработки и функционирования системы. Проводится разработка приложения для работы с базой данных. Раздел включает следующие подразделы:

4.1. Описание информационной модели.

4.2. Общая структура приложения

4.3. Описание особенностей разработки

4.4. Описание компонентов приложения

Информационная модель, распечатка программных модулей и примеры работы приложения приводятся в Приложении.

Заключение

Необходимо привести задачи, которые были решены в процессе выполнения курсовой работы.

Список использованных источников

Приложение

Приложение А

Приложение Б

...

II. Распечатанный титульный лист.

III. Распечатанное задание на курсовую работу.

IV. Распечатанный информационно-удостоверяющий лист.

V. Распечатанный лист с перечнем документов для сдачи в архив.

Все распечатанные документы должны быть подписаны студентом.

Все документы вместе с диском вкладываются в конверт размера А5 с наклеенной этикеткой.

7.3. ПРИМЕРЫ ОФОРМЛЕНИЯ ПРИЛОЖЕНИЙ

В данном разделе приводятся примеры оформления обязательных приложений для курсовой работы по дисциплине «Разработка информационного обеспечения»:

- Даталогическая модель (рис. 39);
- Структурная схема (рис. 40);
- Функциональная схема (рис. 41);
- Информационная модель (рис. 42);
- Примеры работы приложения (рис. 43).

ПРИЛОЖЕНИЕ Д
(обязательное)
Пример работы программы

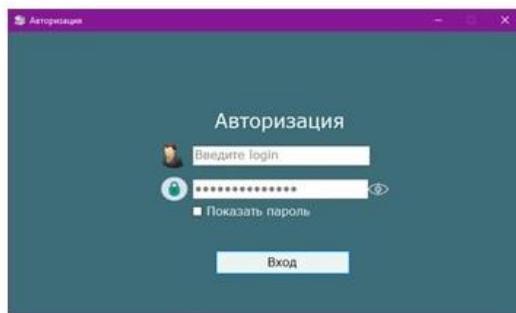


Рис 1. Окно авторизации.

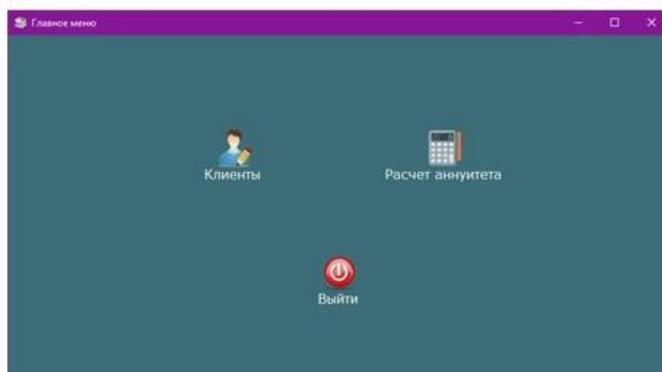


Рис 2. Окно главного меню.

						ТГТУ.09.03.01. NNN КР ТЭ-ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата			19

а)

Рис. 43. Примеры работы программы

а – Приложение; б – продолжение Приложения

ЗАКЛЮЧЕНИЕ

В рамках данного учебного пособия рассмотрены основные подходы к разработке информационного обеспечения.

Приводятся теоретические сведения о базах данных, системах управления базами данных, моделях данных. Подробно рассмотрены основные сведения о реляционных базах данных. Описаны основные этапы проектирования информационного обеспечения.

В пособии даны рекомендации по выполнению курсовой работы по дисциплине «Разработка информационного обеспечения».

Пособие будет полезно студентам бакалавриата и магистратуры направления подготовки «Информатика и вычислительная техника» при изучении дисциплин учебного плана и выполнении выпускной квалификационной работы.

СПИСОК ЛИТЕРАТУРЫ

1. **Дьяков, И. А.** Базы данных. Язык SQL [Электронный ресурс] / И. А. Дьяков : учебное пособие. – Тамбов : Изд-во ТГТУ, 2004.

2. **Майстренко, А. В.** Информационные технологии в науке, образовании и инженерной практике [Электронный ресурс] : учебное пособие / А. В. Майстренко, Н. В. Майстренко. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2014.

3. **Основы** проектирования баз данных в САПР [Электронный ресурс] : учебное пособие / Ю. В. Литовка; Ю. В. Литовка, И. А. Дьяков и др. – Тамбов : Изд-во Тамб. гос. ун-та, 2012.

4. **Давыдова, Д. В.** Базы данных. Основы языка SQL (web-формат) [Электронный ресурс, мультимедиа] : учебное пособие / Д. В. Давыдова, И. Л. Коробова. – Тамбов : Изд-во ФГБОУ ВО «ТГТУ», 2018.

5. **Осипов, Д. Л.** Технологии проектирования баз данных / Д. Л. Осипов. – М. : ДМК Пресс, 2019. – 498 с.

6. **Волк, В. К.** Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. – 2-е изд., стер. – СПб. : Лань, 2021. – 244 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. БАЗЫ ДАННЫХ	5
1.1. Информация и данные	5
1.2. Основные понятия	6
1.3. Банки данных. Требования. Архитектура	7
1.4. Классификация банков данных	10
1.5. Свойства банков данных	11
2. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ	12
2.1. Функции СУБД	12
2.2. Структура СУБД	13
3. МОДЕЛИ ДАННЫХ	15
3.1. Иерархическая модель	17
3.2. Сетевая модель	21
3.3. Достоинства и недостатки иерархических и сетевых СУБД	24
3.4. Реляционная модель	25
3.5. Многомерная модель	26
3.6. Постреляционная модель	29
3.7. Объектно-ориентированная модель	30
4. РЕЛЯЦИОННАЯ БАЗА ДАННЫХ	33
4.1. Таблицы	33
4.2. Первичные ключи	36
4.3. Внешние ключи	36

5. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ	38
5.1. Этапы проектирования баз данных	38
5.2. Инфологическое проектирование	42
5.3. Даталогическое проектирование	48
6. ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ	53
6.1. Дублирование данных	53
6.2. Нормализация отношений	55
7. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ	65
7.1. Цели и задачи курсовой работы	65
7.2. Общие требования к оформлению курсовых работ по дисциплине «Разработка информационного обеспечения»	66
7.3. Примеры оформления приложений	69
ЗАКЛЮЧЕНИЕ	76
СПИСОК ЛИТЕРАТУРЫ	77

Учебное электронное издание

АЛЕКСЕЕВ Сергей Юрьевич
КОРОБОВА Ирина Львовна

РАЗРАБОТКА ИНФОРМАЦИОННОГО
ОБЕСПЕЧЕНИЯ СИСТЕМ
АВТОМАТИЗИРОВАННОЙ ПОДДЕРЖКИ
ПРИНЯТИЯ РЕШЕНИЙ

Учебное пособие

Редактор Л. В. Комбарова
Графический и мультимедийный дизайнер Н. И. Кужильная
Обложка, упаковка, тиражирование Л. В. Комбаровой

ISBN 978-5-8265-2606-4



Подписано к использованию 27.06.2023.
Тираж 50 шт. Заказ № 65

Издательский центр ФГБОУ ВО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14
Телефон (4752) 63-81-08
E-mail: izdatelstvo@tstu.ru