

Учреждение образования  
«Белорусский государственный университет культуры и искусств»  
Факультет информационно-документных коммуникаций  
Кафедра информационных ресурсов и коммуникаций

СОГЛАСОВАНО СОГЛАСОВАНО

Заведующий кафедрой

Декан факультета

\_\_\_\_\_ Ж.Л. Романова

\_\_\_\_\_ Ю.Н. Галковская

« \_\_\_\_ » \_\_\_\_\_ 2020 г. « \_\_\_\_ » \_\_\_\_\_ 2020 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС  
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

**БАЗЫ ДАННЫХ**

специальность

1-23 01 11 Библиотечно-информационная деятельность (по направлениям),  
направление специальности 1-23 01 11-02

Библиотечно-информационная деятельность (автоматизация)

Составитель:

Т.С. Юхновец, старший преподаватель

Рассмотрено и утверждено

на заседании Совета университета 16.06.2020 г.

протокол № 10

Составитель:

*Юхновец Т.С.*, старший преподаватель кафедры информационных ресурсов и коммуникаций учреждения образования «Белорусский государственный университет культуры и искусств»;

Рецензенты:

*Ученый Совет* государственного учреждения «Центральная научная библиотека имени Якуба Коласа Национальной академии наук Беларуси»;

*Н.А. Яцевич*, заведующий кафедрой библиотечно-информационной деятельности, кандидат педагогических наук, доцент

Рассмотрен и рекомендован к утверждению:

*Кафедрой информационных ресурсов и коммуникаций* учреждения образования «Белорусский государственный университет культуры и искусств» (протокол от г. № );

*Советом факультета информационно-документных коммуникаций* учреждения образования «Белорусский государственный университет культуры и искусств» (протокол от г. № ).

## СОДЕРЖАНИЕ

<b>1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА</b> .....	4
<b>2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ</b> .....	7
2.1 Конспект лекций.....	7
<b>3. ПРАКТИЧЕСКИЙ РАЗДЕЛ</b> .....	82
3.1 Методические указания к практическим и лабораторным занятиям .....	82
3.2 Тематика практических и лабораторных занятий .....	83
3.3 Методические указания к семинарским занятиям.....	125
3.4 Тематика семинарских занятий .....	125
<b>4. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ</b> .....	130
4.1 Методические указания к самостоятельной работе .....	131
4.2 Тематика самостоятельной работы .....	131
4.3 График контроля за самостоятельной работой.....	160
4.4 Вопросы к зачету.....	161
4.5 Вопросы к экзамену .....	162
4.6. Тестовые задания .....	165
4.7 Перечень рекомендуемых средств диагностики результатов учебной деятельности студентов.....	174
<b>5. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ</b> .....	175
5.1 Учебная программа.....	175
5.2 Учебно-методические карты учебной дисциплины для дневной и заочной форм получения высшего образования.....	184
5.3 Основная литература .....	191
5.4 Дополнительная литература.....	193

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В современном мире любой вид деятельности, как правило, связан с большим объемом информации. В свою очередь, существует необходимость хранения столь значительных информационных массивов и быстрого поиска в них необходимых сведений. Существенно облегчить эти задачи позволяют базы данных – специальным образом организованная совокупность материалов, использование которой позволяет осуществлять поиск, обработку и извлечение данных, необходимых для удовлетворения информационных потребностей и запросов. Для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации созданы специальные программы – системы управления базами данных.

В деятельности традиционных библиотек как социальных институтов, предназначенных для сбора, обработки, хранения и предоставления информационных ресурсов, базы данных занимают одно из приоритетных мест и имеют стратегическое значение, поскольку оказывают определяющее влияние на ее эффективность. Каждая библиотека уже столкнулась или в самое ближайшее время столкнется с проблемами разработки, формирования и использования баз данных. От уровня знаний баз данных и технологий создания и поддержки их функциональности, которыми владеют библиотечные специалисты, зависит качество информационного обслуживания в библиотеках. В связи с этим особую актуальность приобретает освоение библиотечными специалистами принципов создания и эффективного применения баз данных и систем управления базами данных, средств администрирования и защиты баз данных.

Учебная дисциплина «Базы данных» является одной из основных специальных дисциплин при подготовке студентов по направлению специальности 1-23 01 11-02 «Библиотечно-информационная деятельность (автоматизация)», которая входит в обязательный компонент учебного плана.

Учебная дисциплина «Базы данных» является одним из компонентов комплекса учебных дисциплин, направленных на подготовку высококвалифицированных библиотечно-информационных специалистов в области формирования и использования информационных ресурсов.

Данный учебно-методический комплекс (далее – УМК) представляет собой систему дидактических средств обучения по учебной дисциплине «Базы данных». Он является структурно-логической моделью процесса формирования профессионально значимых компетенций библиотекаря-

библиографа, необходимых для понимания сущности, классификации, функций и особенностей баз данных и систем управления базами данных; проектирования, создания и использования баз данных библиотек; обеспечения безопасности баз данных и их функционирования с учетом нормативной правовой базы охраны авторских прав.

*Цель УМК* по учебной дисциплине «Базы данных» – это систематизация учебно-методических материалов, необходимых при изучении теоретических, организационных, технологических вопросов создания, функционирования и использования баз данных в условиях деятельности библиотек; учебно-методическая поддержка студентов в освоении учебного материала; повышение эффективности организации учебного процесса и самостоятельной работы студентов на основе компетентностного подхода.

*Задачи УМК:*

- систематизация содержания учебной дисциплины «Базы данных»;
- упорядочение процесса изучения учебной дисциплины с учетом достижений науки и практики;
- обеспечение организации самостоятельной учебной работы и контроля знаний студентов;
- оказание студентам методической помощи в освоении учебного материала;
- оказание преподавателям методической помощи, необходимой для качественного преподавания данной учебной дисциплины.

Структура УМК соответствует рекомендациям «Положения об учебно-методическом комплексе в учреждении образования «Белорусский государственный университет культуры и искусств» (утверждено Приказом ректора университета 26.04.2017, № 69) и включает в себя следующие разделы:

Теоретический раздел. В соответствии с учебной программой по курсу все разделы и темы представлены краткими текстами лекций в данном УМК.

Практический раздел – содержит рабочие материалы, задания к практическим занятиям, методические рекомендации к их выполнению в объеме, определенном учебной программой, которые будут способствовать усвоению, формированию умений и навыков в области теории и практики автоматизации библиотечно-информационной деятельности, закреплению теоретических знаний и их проверке. В раздел включается также тематика семинарских занятий, порядок их проведения, индивидуальные задания для подготовки, рекомендуемая литература для подготовки к занятиям и закрепления пройденного материала.

Раздел контроля знаний – включает задания для самостоятельной и контролируемой самостоятельной работы, тестовые задания по темам учебной дисциплины, перечни контрольных вопросов к зачету и экзамену, перечень рекомендованных средств диагностики результатов учебной деятельности студентов.

Вспомогательный раздел включает учебную программу по учебной дисциплине, учебно-методическую карту учебной дисциплины для дневной и заочной форм получения образования, списки основной и дополнительной литературы.

РЕПОЗИТОРИЙ БГУКИ

## 2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

### 2.1 Конспект лекций

#### Тема 1. Основы теории баз данных

##### План:

1. Основные понятия теории баз данных.
2. Банк данных как информационная система.
3. База данных как интегрированный ресурс.
4. Категории пользователей банков и баз данных.

1. Основные понятия теории баз данных.

Одним из важнейших понятий в теории баз данных является понятие *информации*. Под *информацией* понимаются любые сведения о каком-либо событии, процессе, объекте.

*Данные* – это информация, представленная в определенном виде, позволяющем автоматизировать ее сбор, хранение и дальнейшую обработку человеком или информационным средством. Для компьютерных технологий данные – это информация в дискретном, фиксированном виде, удобная для хранения, обработки на компьютерах, а также для передачи по каналам связи.

*База данных* (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области, или иначе БД – это совокупность взаимосвязанных данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений в определенной предметной области. БД состоит из множества связанных файлов.

Отличительной чертой баз данных следует считать то, что данные хранятся совместно с их описанием, а в прикладных программах описание данных не содержится. Независимые от программ пользователя данные обычно называются *метаданными*. В ряде современных систем метаданные, содержащие также информацию о пользователях, форматы отображения, статистику обращения к данным и другие сведения, хранятся в *словаре базы данных*.

*Система управления базами данных (СУБД)* – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Для облегчения обработки информации создаются *информационные системы (ИС)*. Большинство существующих ИС являются автоматизированными.

*Автоматизированная информационная система (АИС)* – это система, реализующая автоматизированный сбор, обработку, манипулирование данными, функционирующая на основе компьютеров и других технических средств и включающая соответствующее программное обеспечение (ПО) и персонал. В этом качестве для краткости может использоваться термин «*информационная система*» (ИС), который подразумевает понятие «автоматизированная информационная система».

В широком понимании под определение ИС подпадает любая система обработки информации. По области применения ИС можно разделить на системы, используемые в производстве, образовании, здравоохранении, науке, военном деле, социальной сфере, торговле и других отраслях. По целевой функции ИС можно условно разделить на следующие основные категории: управляющие, информационно-справочные, поддержки принятия решений.

Иногда используется более узкая трактовка понятия ИС как совокупности аппаратно-программных средств, задействованных для решения некоторой прикладной задачи.

Информационные системы имеют следующие особенности:

- для обеспечения их работы нужны сравнительно низкие вычислительные мощности;
- данные, которые они используют, имеют сложную структуру;
- необходимы средства сохранения данных между последовательными запусками системы.

Каждая ИС в зависимости от ее назначения имеет дело с той или иной частью реального мира, которую принято называть *предметной областью (ПрО) системы*. Выявление ПрО – это необходимый начальный этап разработки любой ИС. Именно на этом этапе определяются информационные потребности всей совокупности пользователей будущей системы, которые, в свою очередь, определяют содержание ее базы данных.

*Банк данных (БД)* является разновидностью ИС. БД – это система специальным образом организованных данных: баз данных,

программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

Отдельные программы или комплекс программ, реализующие автоматизацию решения прикладных задач обработки данных, называются *приложениями*. Приложения, созданные средствами СУБД, относятся к *приложениям СУБД*. Приложения, созданные вне среды СУБД с помощью систем программирования, использующих средства доступа к БД, например, Delphi или VisualStudio, называют *внешними приложениями*. Для работы с БД зачастую достаточно средств СУБД и не нужно использовать приложения, создание которых обычно требует программирования. Приложения разрабатывают главным образом в случаях, когда требуется сделать работу пользователей более удобной или автоматизировать рутинные операции с БД.

## 2. Банк данных как информационная система.

Существует множество определений банка данных. Например, в «Толковом словаре по вычислительным системам» дается следующее определение: «Банк данных – это система, предоставляющая услуги по хранению и поиску данных определенной группе пользователей и по определенной тематике (например, биологические виды, статистика торговли, цены на товары). При этом «сами данные могут быть организованы в виде базы данных или одного или более файлов». Определение банка данных, опубликованное в «Общетраслевых руководящих материалах по созданию банков данных Государственного комитета по науке и технике» (ГКНТ): «Банк данных – это система специальным образом организованных данных (баз данных), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных» (цит. по: Любицкий, Ю. В. Базы данных [Текст] : учебное пособие / Мин-во образования и науки Рос. Федерации, Федер. агентство по образованию, Хабаровская госуд. академия экономики и права, Каф. информац. технологий. – Хабаровск : РИЦ ХГАЭП, 2005. – 80 с., С. 6). В данном определении подчеркивается, что банк данных является сложной системой, включающей в себя всеобеспечивающие подсистемы, необходимые для функционирования любой системы автоматизированной обработки данных.

А.Б. Антопольский определяет БД как «автоматизированную информационную систему централизованного хранения и коллективного использования данных»

В законе Республики Беларусь «Об информации, информатизации и защите информации» БД определяется как «организационно-техническая система, включающая одну или несколько баз данных и систему управления ими».

Как правило, БД являются системами коллективного пользования. К информации, хранимой в них, часто можно получить доступ по телекоммуникационным сетям.

*Отличительные особенности банков данных (БД):*

1. БД создаются обычно не для решения какой-либо одной задачи для одного пользователя, а для многоцелевого использования. Они отражают определенную часть реального мира. Эта информация должна по возможности фиксироваться в БД однократно, и все пользователи, которым эта информация нужна, должны иметь возможность работать с ней.

2. Наличие в БД специальных языковых и программных средств, облегчающих для пользователей выполнение всех операций, связанных с организацией хранения данных, их корректировки и доступа к ним.

Термин «банк данных» является общепризнанным. В некоторой англоязычной литературе используется термин «система баз данных» (databasesystem), который по своему содержанию близок введенному понятию банка данных (система баз данных включает базу данных, систему управления базами данных, соответствующее оборудование и персонал).

*Основные требования, предъявляемые к БД:*

- адекватность отображения предметной области (полнота, целостности и непротиворечивости данных, актуальность информации, т.е. ее соответствие состоянию объекта на данный момент времени);
- возможность взаимодействия пользователей разных категорий и в разных режимах, обеспечение высокой эффективности доступа для разных приложений;
- дружелюбность интерфейсов и малое время на освоение системы, особенно для конечных пользователей;
- обеспечение секретности и конфиденциальности для некоторой части данных;
- определение групп пользователей и их полномочий;
- обеспечение взаимной независимости программ и данных;
- обеспечение надежности функционирования БД;

- защита данных от случайного и преднамеренного разрушения;
- возможность быстрого и полного восстановления данных в случае их разрушения;
- технологичность обработки данных, приемлемые характеристики функционирования БнД (стоимость обработки, время реакции системы на запросы, требуемые машинные ресурсы и др.).

*Основными функциями БнД являются:*

1. Хранение информации, ее защита и восстановление после сбоев в работе.
2. Периодическое изменение хранимых данных.
3. Поиск и отбор необходимых данных по запросам пользователей и прикладных программ.
4. Обработка найденных данных и вывод результатов в заданной форме.

БнД является сложной человеко-машинной системой, включающей в свой состав различные взаимосвязанные и взаимозависимые компоненты (рис.1.1).



Рис. 1.1 Компоненты банка данных

*Информационная компонента.* Ядром БнД является база данных. База данных—это поименованная совокупность взаимосвязанных данных, находящихся под управлением системы управления базами данных (СУБД).

Описания баз данных относятся к метаинформации, т. е. информации об информации. Описание баз данных часто называют *схемой*. Кроме того, в БнД могут присутствовать описания отдельных частей баз данных с точки

зрения отдельных пользователей — подсхемы.

Кроме описания баз данных в состав метаинформации, хранимой в БД, может включаться информация о предметной области, необходимая для проектирования системы, о пользователях БД, о проектных решениях и некоторая другая информация.

Централизованное хранилище метаинформации называется *словарем данных*. В литературе используются также термины словарь-справочник, энциклопедия, репозиторий.

*Программные средства*. Программные средства БД представляют собой сложный комплекс, обеспечивающий взаимодействие всех частей информационной системы при ее функционировании (рис. 1.2).

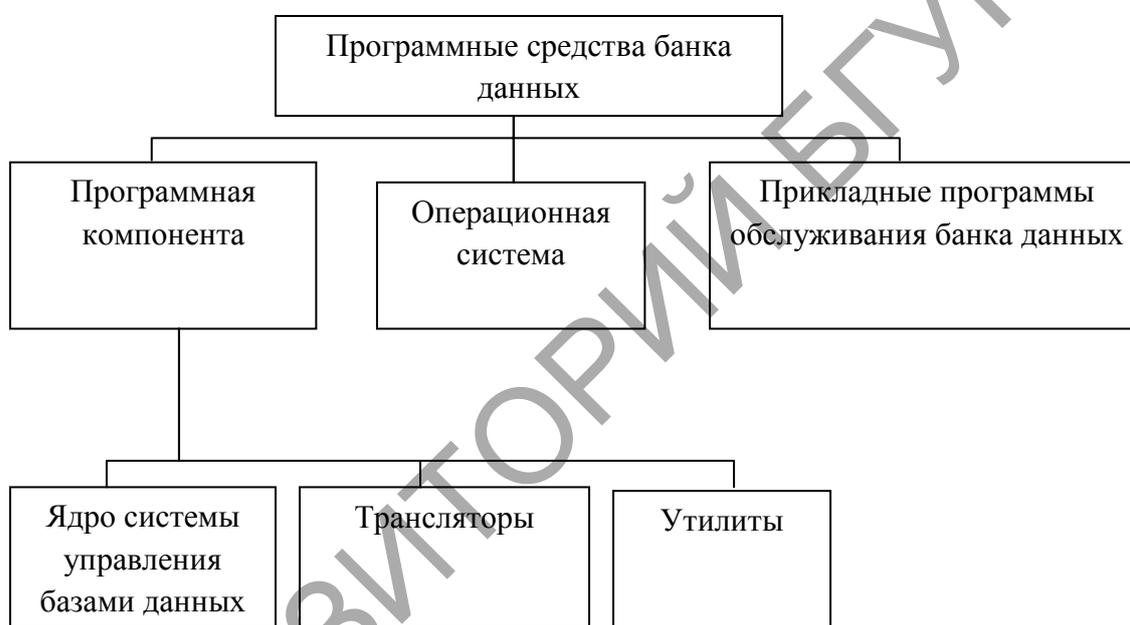


Рис. 1.2 Программные средства банка данных

Основу программных средств БД представляет СУБД. В ней можно выделить ядро СУБД, обеспечивающее организацию ввода, обработки и хранения данных, а также другие компоненты, обеспечивающие настройку системы, средства тестирования, утилиты, обеспечивающие выполнение вспомогательных функций, таких, как восстановление баз данных, сбор статистики о функционировании БД и др. Важной компонентой СУБД являются трансляторы или компиляторы для используемых ею языковых средств. Утилиты обеспечивают выполнение вспомогательных функций (например, ведение журнала статистики работы системы и др.).

подавляющее большинство СУБД работает в среде универсальных операционных систем и взаимодействует с операционной системой при обработке обращений к БД. Поэтому можно считать, что операционная

система также входит в состав БнД.

Для обработки запросов к БнД пишутся соответствующие программы, которые представляют прикладное программное обеспечение БнД.

*Языковые средства.* При проектировании языков четвертого поколения используются следующие принципы:

1. Принцип минимума работы: язык должен обеспечить минимум усилий, чтобы привести БнД в рабочее состояние.

2. Принцип минимума мастерства: работа с языком должна быть по возможности проста.

3. Принцип естественности языка, упразднения «инородного» синтаксиса и мнемоники. Язык не должен требовать от пользователей значительных усилий в изучении синтаксиса или содержать много мнемонических или иных обозначений, которые быстро забываются.

4. Принцип минимума времени. Язык должен позволять без существенной задержки реализовывать возникающие потребности в доступе к информации и ее обработке.

5. Принцип минимума ошибок. Технология должна быть спроектирована таким образом, чтобы минимизировать ошибки человека.

6. Принцип минимума поддержки. Механизм языков четвертого поколения должен позволить легко вносить изменения в имеющиеся приложения.

7. Принцип максимума результата. Языки четвертого поколения предоставляют пользователям мощный инструмент для решения разнообразных задач.

*Организационно-методические средства.* Организационно-методические средства БнД представляют собой различные инструкции, методические и регламентирующие материалы, предназначенные для пользователей разных категорий, взаимодействующих с БнД.

*Администраторы.* Функционирование БнД невозможно без участия специалистов, обеспечивающих создание, функционирование и развитие БнД. Такая группа специалистов называется администратором банка данных (АБД). Эта группа специалистов считается составной частью банка данных.

БнД являются сложными системами, и их классификация может быть произведена по разным признакам. Одни из признаков классификации и соответственно классификационные группировки относятся к БнД в целом, другие – к отдельным его компонентам, третьи могут быть отнесены как к отдельному компоненту, так и к нескольким компонентам или БнД в целом.

*Преимущества БнД:*

1. Наличие единого отображения определенной части реального мира позволяет обеспечить непротиворечивость и целостность информации, возможность обращаться к ней не только при решении заранее определенных задач, но и с нерегламентированными запросами.

2. Интегрированное хранение сокращает избыточность хранимых данных, что приводит к сокращению затрат не только на создание и хранение данных, но и на поддержание их в актуальном состоянии.

3. Использование БД при правильной его организации должно существенно изменить деятельность организации, где он внедряется: привести к обеспечению большей доступности данных для всех категорий сотрудников, сокращению документооборота, возможности получения разнообразных по форме и содержанию документов, перераспределению функций между сотрудниками и изменению характера выполняемых функций и, как следствие, улучшить всю систему управления организацией.

4. Использование СУБД обеспечивает высокое качество выполнения функций по управлению данными и облегчает процесс создания информационных систем (ИС).

5. Выделение специальной группы сотрудников, выполняющих функции по проектированию и развитию БД (администраторов БД), и освобождение от этих функций всех остальных пользователей не только приводит к снижению требований к остальным участникам процесса создания и функционирования БД, но и повышает качество разработок, так как вопросами организации данных занимается небольшое число профессионалов в этой области.

6. БД обеспечивают возможность более полной реализации принципа независимости прикладных программ от данных, чем это возможно при организации локальных файлов.

#### *Недостатки БД.*

Недостатки БД вытекают из их достоинств. Создание интегрированной системы сложнее, чем создание множества локальных систем. Как следствие, предъявляются высокие требования к квалификации разработчиков БД. В результате интеграции возможна некоторая потеря эффективности отдельных приложений (но общая эффективность всей системы будет выше). Для управления данными требуется специализированное программное обеспечение, которое, в зависимости от класса системы, может быть сравнительно дорогим, предъявляющим повышенные требования к техническим средствам. Эксплуатация распределенных корпоративных БД – процесс сложный и дорогостоящий. Правильный выбор СУБД позволит

свести отрицательные последствия к минимуму.

### 3. База данных как интегрированный ресурс.

Исторически понятие базы данных возникло как альтернатива файловой организации данных при хранении с помощью ЭВМ (на магнитных носителях). Такая организация данных была характерна для прикладного программного обеспечения на начальном этапе распространения вычислительной техники. Файловая организация предполагала хранение данных в виде совокупности файлов, ориентированных на использование какой-либо одной прикладной программы, предназначенной для решения некоторой специфической задачи. Такая неуниверсальность в организации информации привела к большой избыточности (дублированию) при хранении, противоречивости данных, хранящихся в различных системах.

Понятие «база данных» возникло в результате стандартизации и унификации данных, универсально организованных и хранящихся с помощью ЭВМ с целью использования для многих приложений. При этом описание данных уже не скрыто в программах, а явным образом декларируется и хранится в самой базе. База данных может быть определена как структурная совокупность данных, поддерживаемых в актуальном состоянии (в соответствии объектам некоторой предметной области) и служащая для удовлетворения информационных потребностей многих пользователей. Для поддержания актуальности данных, хранящихся в базе, получения сведений об информационных запросах, перехода к данным и программам пользователей служат СУБД. Более подробно СУБД будут рассматриваться в следующей лекции.

Согласно ГОСТ 20886–85 «Организация данных в системах обработки данных. Термины и определения» БД – это «совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ».

В ГОСТ Р ИСО/МЭК ТО 10032–2007 «Эталонная модель управления данными» БД определяется как «совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных».

А.Б. Антопольский определяет БД как «объективную форму представления и организации совокупности данных (статей и др.), систематизированных таким образом, чтобы эти данные могли быть найдены

и обработаны ЭВМ».

Согласно статье 1260 Гражданского кодекса Российской Федерации «базой данных является представленная в объективной форме совокупность самостоятельных материалов (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ)»

В законе Республики Беларусь «Об информации, информатизации и защите информации» БД определяется как «совокупность структурированной и взаимосвязанной информации, организованной по определенным правилам на материальных носителях».

В БД информация должна быть организована так, чтобы обеспечить минимальную долю ее избыточности. Частичная избыточность информации необходима, но она должна быть минимизирована, так как чрезмерная избыточность данных влечет за собой ряд негативных последствий. Главные из них:

- увеличение объема информации, а значит, потребность в дополнительных ресурсах для хранения и обработки дополнительных объемов данных;
- появление ошибок при вводе дублирующей информации, нарушающих целостность БД и создающих противоречивые данные.

В БД должны храниться данные, логически связанные между собой. Для того, чтобы данные можно было связать между собой, и связать так, чтобы эти связи соответствовали реально существующим в данной предметной области, последнюю подвергают детальному анализу, выделяя сущности или объекты. Сущность или объект – это то, о чем необходимо хранить информацию. Сущности имеют некоторые характеристики, называемые атрибутами, которые тоже необходимо сохранять в БД. Атрибуты по своей внутренней структуре могут быть простыми, а могут быть сложными. Простые атрибуты могут быть представлены простыми типами данных. Различного рода графические изображения, являющиеся атрибутами сущностей, – это пример сложного атрибута. Определив сущности и их атрибуты, необходимо перейти к выявлению связей, которые могут

существовать между некоторыми сущностями. Связь – это то, что объединяет две или более сущностей. Связи между сущностями также являются частью данных, и они также должны храниться в базе данных.

Перечень важнейших *требований*, которым должны удовлетворять

современные БД:

- высокое быстродействие (малое время отклика на запрос);
- простота обновления данных;
- независимость данных;
- интегрированность данных;
- целостность данных;
- динамичность данных и способность к расширению;
- гибкость и адаптивность структуры БД;
- совместное использование данных многими пользователями;
- безопасность данных – защита данных от преднамеренного или непреднамеренного нарушения секретности, искажения или разрушения.
- стандартизация построения и эксплуатации БД (фактически СУБД).
- адекватность отображения данных соответствующей предметной области;
  - возможность поиска по многим ключам.

Важнейшими требованиями являются первые два противоречивых требования: повышение быстродействия требует упрощения структуры БД, что, в свою очередь, затрудняет процедуру обновления данных, увеличивает их избыточность.

Независимость данных – возможность изменения логической и физической структуры БД без изменения представлений пользователей. Независимость данных предполагает инвариантность к характеру хранения данных, программному обеспечению и техническим средствам. Она обеспечивает минимальные изменения структуры БД при изменениях стратегии доступа к данным и структуры самих исходных данных.

Безопасность данных включает их целостность и защиту. Целостность данных – устойчивость хранимых данных к разрушению и уничтожению, связанных с неисправностями технических средств, системными ошибками и ошибочными действиями пользователей. Она предполагает:

- отсутствие неточно введенных данных или двух одинаковых записей об одном и том же факте;
- защиту от ошибок при обновлении БД;
- невозможность удаления (или каскадное удаление) связанных данных разных таблиц;
- отсутствие искажения данных при работе в многопользовательском режиме и в распределенных базах данных;
- сохранность данных при сбоях техники (восстановление данных).

Целостность обеспечивается триггерами целостности – специальными

приложениями-программами, работающими при определенных условиях.

Защита данных от несанкционированного доступа предполагает ограничение доступа к конфиденциальным данным и может достигаться:

- введением системы паролей;
- получением разрешений от администратора базы данных (АБД);
- запретом от АБД на доступ к данным;

Стандартизация обеспечивает преемственность поколений СУБД, упрощает взаимодействие БД одного поколения СУБД с одинаковыми и различными моделями данных. Стандартизация (ANSI/SPARC) осуществлена в значительной степени в части интерфейса пользователя СУБД и языка SQL. Это позволило успешно решить задачу взаимодействия различных реляционных СУБД с помощью языка SQL.

На уровне информационных систем БД рассматривается как компонент, представляющий собой информационную модель предметной области. Здесь наиболее важной является проблема логической структуры БД.

При рассмотрении БД на уровне информационных ресурсов БД трактуется как элемент мировых информационных ресурсов. Основной характеристикой здесь является содержание БД, хотя и структуры данных также немаловажны.

Классификация БД по основным признакам приведена на рис. 1.3.

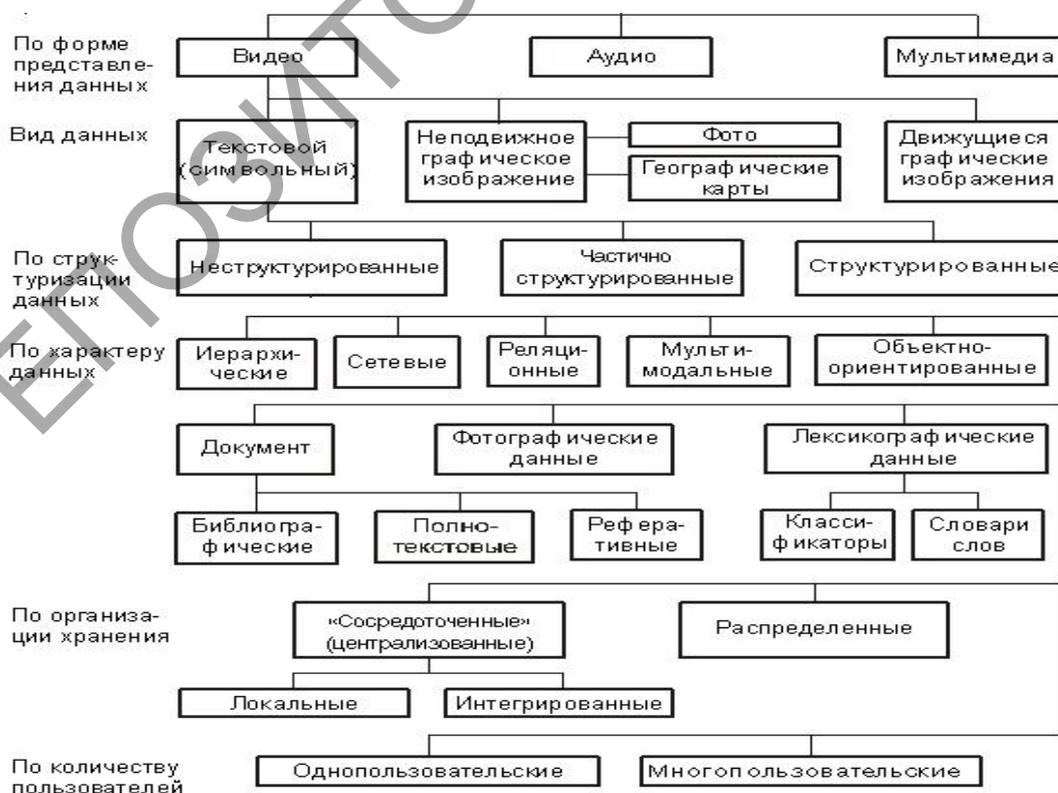


Рис. 1.3 Классификация баз данных по основным признакам

#### 4. Категории пользователей банков и баз данных.

С БнДи БД в процессе их создания и функционирования взаимодействуют пользователи разных категорий. Основной категорией пользователей являются так называемые *конечные пользователи*, т.е. те пользователи, для нужд которых и создаются БнД и БД. В зависимости от особенностей создаваемых БнД и БД круг их конечных пользователей может существенно различаться. Это могут быть случайные пользователи, обращающиеся к ним время от времени, а могут быть регулярные пользователи. Конечные пользователи могут отличаться друг от друга и степенью владения компьютерной техникой. От конечных пользователей не должно требоваться каких-то специальных знаний в области вычислительной техники и языковых средств.

*Администраторы БнД и БД* также являются их специфическими пользователями. Обычно они обращаются к БнД не за информацией о предметной области, а к метаинформации, а также используют ресурсы БнД для выполнения своих функций. В зависимости от сложности и объема БнД данных, от особенностей используемой СУБД служба администрации БнД данных может различаться как по составу и квалификации специалистов, так и по количеству работающих в этой службе.

В составе группы администраторов БнД можно выделить различные подгруппы в зависимости от выполняемых ими функций. Численность группы администрации, выполняемые ими функции будут в некоторой степени зависеть от масштабов БнД и БД, специфики хранимой в них информации, особенностей используемых программных средств и некоторых других факторов. В составе администрации БнД должны быть системные аналитики, проектировщики структур данных и внешнего по отношению к банку данных информационного обеспечения, проектировщики технологических процессов обработки данных, системные и прикладные программисты, операторы, специалисты по техническому обслуживанию.

Третьей категорией пользователей БнД и БД являются *разработчики и администраторы приложений*. Эта группа пользователей функционирует во время проектирования, создания и реорганизации БнД. Администраторы приложений координируют работу разработчиков при разработке конкретного приложения или группы приложений, объединенных в функциональную подсистему. Разработчики конкретных приложений

работают с той частью информации из БД, которая требуется для конкретного приложения.

Не для каждого БД могут быть выделены все категории пользователей. При разработке БД с использованием настольных СУБД администратор БД, администратор приложений и разработчик часто существовали в одном лице. Однако при построении современных сложных корпоративных БД данных могут существовать и группы администраторов приложений, и отделы разработчиков. Наиболее сложные обязанности возложены на группу администратора БД.

Администраторы БД и БД выполняют большой круг разнообразных функции.

*Связи администратора банка данных.* В процессе своей деятельности администратор БД взаимодействует с другими категориями пользователей банка данных, а также и с «внешними» специалистами, не являющимися пользователями БД (рис. 1.4).



Рисунок 1.4. Взаимодействие администратора БД с другими категориями пользователей БД

Прежде всего, если БД создается для информационного обслуживания какого-либо предприятия или организации, необходимы контакты с администрацией этой организации. Внедрение БД приводит к большим изменениям не только системы обработки данных, но и всей системы управления организацией. Руководство организации должно быть ознакомлено с возможностями, предоставляемыми БД, проинформировано

об их преимуществах и недостатках, а также о проблемах, вызываемых созданием и функционированием БД.

Поскольку БД является динамическим информационным отображением предметной области, то желательно, чтобы администратор БД, в свою очередь, был своевременно информирован о перспективах развития объекта, для которого создается БД.

Руководством организации и администратором БД должны быть согласованы цели, основные направления и сроки создания БД и его развития, очередность подключения пользователей.

Очень тесная связь у администратора БД на всех этапах жизненного цикла БД наблюдается с конечными пользователями. Это взаимодействие возникает на начальных стадиях проектирования системы, когда изучаются потребности пользователей, уточняются особенности предметной области, и постоянно поддерживается в процессе проектирования и функционирования системы.

Следует отметить, что в последнее время наблюдается активное перераспределение функций между конечными пользователями и администратором БД. Это, прежде всего, связано с развитием языковых и программных средств, ориентированных на конечных пользователей. К ним относятся простые и одновременно мощные языки запросов, а также средства автоматизации проектирования.

Если БД функционирует в составе какой-либо автоматизированной информационной системы, то администратор БД должен работать в контакте со специалистами по обработке данных в этой системе.

Администраторы БД взаимодействуют также с внешними по отношению к нему группами специалистов, и, прежде всего, поставщиками СУБД и прикладными программистами, администраторами других БД. (Диго, С.М. Базы данных: проектирование и использование : учебник. – М. : Финансы и статистика, 2005. – 592 с. : ил.).

## **Тема 2. Системы управления базами данных**

### **План:**

1. Основные подходы к определению понятия «система управления базами данных». Возможности и функции систем управления базами данных.
2. Критерии выбора и структура систем управления базами данных.
3. Классификация систем управления базами данных.

1. Основные подходы к определению понятия «система управления базами данных». Возможности и функции систем управления базами данных.

Определяя базы данных как среду хранения данных, отделенную от процедур, следует подразумевать наличие специальных программных средств доступа к данным – систем управления базами данных (СУБД). СУБД определяется по-разному. Приведем некоторые определения:

1. Согласно ГОСТ 20886–85 «Организация данных в системах обработки данных. Термины и определения» СУБД – это «совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами».

2. Согласно ГОСТ Р ИСО/МЭК ТО 10032–2007 «Эталонная модель управления данными» СУБД – это «Совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных».

3. С.Д. Кузнецов определяет СУБД как «комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать)».

За время развития БД и систем их управления было создано множество различных СУБД. Многие из них уже устарели технически и заменены новыми продуктами. Тем не менее, можно разделить все СУБД на две глобальные категории: *СУБД общего назначения* и *специализированные СУБД*. Как исходит из названия, при разработке СУБД общего назначения не делается акцент на какой-либо стороне работы СУБД, так же как программа не создается только для некоего узкого круга БД. СУБД общего назначения являются многофункциональными продуктами, которые уже в ходе создания либо эксплуатации БД могут быть модифицированы под нужды пользователя либо администратора системы. В связи с широкими возможностями СУБД общего назначения, часто это коммерческие продукты, поставляемые пользователям, которые в дальнейшем могут использовать в полной мере те специфические особенности данной СУБД, которые им нужны. Такие СУБД обладают средствами настройки на работу с конкретной БД. Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности.

Специализированные СУБД создаются в редких случаях при

невозможности или нецелесообразности использования СУБД общего назначения.

СУБД общего назначения – это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией БД.

Рынок программного обеспечения персональных компьютеров располагает большим числом разнообразных по своим функциональным возможностям коммерческих СУБД общего назначения, а также средствами их окружения практически для всех массовых моделей машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и надежной безопасности, что дает возможность разработчикам гарантировать полную безопасность данных при меньших затратах сил на низкоуровневое программирование. Продукты, функционирующие в среде WINDOWS, выгодно отличаются удобством пользовательского интерфейса и встроенными средствами повышения производительности.

Приведем сравнение СУБД ACCESS, MySQL и Oracle по некоторым параметрам:

1. Объем памяти на жестком диске, необходимый для самой СУБД: ACCESS (OfficeXP) – 530 Мб, Oracle – > 1 Гб, для работы с MySQL + PHP через Интернет-сервер необходим только браузер (например Internet Explorer – 14,7 Мб), а для работы локально нужен еще сервер, поддерживающий MySQL и PHP (например Apache – 8 Мб).

2. Размер БД в формате, соответствующем каждой СУБД: ACCESS – 1,73 Мб, MySQL – 113 Кб, Oracle – размер определяется не содержанием самой базы, а задаваемым табличным пространством.

3. Оперативная память, используемая СУБД при работе с БД: ACCESS – 4528 Кб, сервер Apache + Internet Explorer – 28 612 Кб (из них Internet Explorer – 11 660 Кб).

4. Быстродействие: при работе локально разница между временем выполнения запроса в ACCESS и временем выполнения аналогичного запроса в MySQL (Internet Explorer – Apache – Internet Explorer) практически неощутима (десятые доли секунды); при работе же с MySQL через Internet скорость зависит от таких параметров, как трафик сети, удаленность и быстродействие сервера и пр.

5. Простота использования: ACCESS, как и все продукты из MS OFFICE, очень наглядна, содержит хорошую систему помощи и опции так

называемых «мастеров» создания и заполнения. Это все в совокупности позволяет даже неопытному пользователю, не имеющему навыков работы с какими-либо СУБД, довольно быстро научиться создавать и управлять своими БД; MySQL – несмотря на то, что приходится прописывать все в ручную, особых трудностей не вызывает, особенно если пользователь обладает хоть какими-то навыками программирования и работы с БД. Oracle – это СУБД несколько иного уровня, и поэтому требует изучения в течение большего, по сравнению с ACCESS и MySQL, времени.

*Производительность СУБД* оценивается:

- временем выполнения запросов;
- скоростью поиска информации в неиндексированных полях;
- временем выполнения операций импортирования базы данных из других форматов;
- скоростью создания индексов и выполнения таких массовых операций, как обновление, вставка, удаление данных;
- максимальным числом параллельных обращений к данным в многопользовательском режиме;
- временем генерации отчета.

На производительность СУБД оказывают влияние два фактора:

1. СУБД, которые следят за соблюдением целостности данных, несут дополнительную нагрузку, которую не испытывают другие программы.
2. Производительность собственных прикладных программ сильно зависит от правильного проектирования и построения БД.

В процессе реализации своих функций СУБД постоянно взаимодействует с БД и с другими прикладными программными продуктами пользователя, предназначенными для работы с данной БД и называемыми приложениями.

Для того, чтобы СУБД успешно справлялась со своими задачами, она должна обладать определенными возможностями.

*Обобщенная характеристика возможностей современных СУБД:*

1. СУБД включает язык определения данных, с помощью которого можно определить БД, ее структуру, типы данных, а также средства задания ограничений для хранимой информации. В многопользовательском варианте СУБД этот язык позволяет формировать представления как некоторое подмножество БД, с поддержкой которых пользователь может создавать свой взгляд на хранимые данные, обеспечивать дополнительный уровень безопасности данных и многое другое.
2. СУБД позволяет вставлять, удалять, обновлять и извлекать

информацию из БД посредством языка управления данными.

3. Большинство СУБД могут работать на компьютерах с разной архитектурой и под разными операционными системами, причем наработку пользователя при доступе к данным практически тип платформы влияния не оказывает.

4. Многопользовательские СУБД имеют достаточно развитые средства администрирования БД.

5. СУБД предоставляет контролируемый доступ к БД с помощью:

– системы обеспечения безопасности, предотвращающей несанкционированный доступ к информации БД;

– системы поддержки целостности БД, обеспечивающей непротиворечивое состояние хранимых данных;

– системы управления параллельной работой приложений, контролирующей процессы их совместного доступа к БД;

– системы восстановления, позволяющей восстановить БД до предыдущего непротиворечивого состояния, нарушенного в результате аппаратного или программного обеспечения.

*Функции СУБД:*

1. Управление данными во внешней памяти. Данная функция предоставляет пользователям возможности выполнения самых основных операций, которые осуществляются с данными, – сохранение, извлечение и обновление информации. Она включает в себя обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например для ускорения доступа к данным.

2. Управление транзакциями. Транзакция – это последовательность операций над БД, рассматриваемых СУБД как единое целое. Транзакция представляет собой набор действий, выполняемых с целью доступа или изменения содержимого БД. Примерами простых транзакций может служить добавление, обновление или удаление в БД сведений о некоем объекте. Сложная же транзакция образуется в том случае, когда в БД требуется внести сразу несколько изменений. Инициализация транзакции может быть вызвана отдельным пользователем или прикладной программой.

3. Восстановление БД. Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида

аппаратных сбоев: 1) мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания); 2) жесткие сбои, характеризуемые потерей информации на носителях внешней памяти. Поддержание надежности хранения данных в БД требует избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД.

4. Поддержка языков БД. Для работы с БД используются специальные языки, называемые языками БД. В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с БД. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language – язык структурированных запросов). Язык SQL позволяет определять схему реляционной БД и манипулировать данными.

5. Поддержка словаря данных. Наличие интегрированного системного каталога с данными о схемах, пользователях, приложениях и т. д. Системный каталог, который еще называют словарем данных, является, таким образом, хранилищем информации, описывающей данные в БД. Предполагается, что каталог доступен как пользователям, так и функциям СУБД. Обычно в словаре данных содержится следующая информация: имена, типы и размеры элементов данных; имена связей; накладываемые на данные ограничения поддержки целостности; имена пользователей, которым предоставлено право доступа к данным; внешняя, концептуальная и внутренняя схемы и отображения между ними; статистические данные, например частота транзакций и счетчики обращений к объектам БД.

6. Управление параллельным доступом. Одна из основных целей создания и использования СУБД заключается в том, чтобы множество пользователей могло осуществлять параллельный доступ к совместно обрабатываемым данным. Параллельный доступ сравнительно просто организовать, если все пользователи выполняют только чтение данных, поскольку в этом случае они не могут помешать друг другу. Однако когда два или больше пользователей одновременно получают доступ к БД, конфликт с нежелательными последствиями легко может возникнуть, например, если хотя бы один из них попытается обновить данные. СУБД должна гарантировать, что при одновременном доступе к БД многих

пользователей подобных конфликтов не произойдет.

7. Управление буферами оперативной памяти. СУБД обычно работают с БД значительного размера. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. В развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов.

8. Контроль доступа к данным. СУБД должна иметь механизм, гарантирующий возможность доступа к БД только санкционированных пользователей и защищающий ее от любого несанкционированного доступа. В современных СУБД поддерживается один из двух широко распространенных подходов к вопросу обеспечения безопасности данных: избирательный подход или обязательный подход. В большинстве современных систем предусматривается избирательный подход, при котором некий пользователь обладает различными правами при работе с разными объектами. Значительно реже применяется альтернативный, обязательный подход, где каждому объекту данных присваивается некоторый классификационный уровень, а каждый пользователь обладает некоторым уровнем допуска.

9. Поддержка целостности данных. Термин «Целостность» используется для описания корректности и непротиворечивости хранимых в БД данных. Реализация поддержки целостности данных предполагает, что СУБД должна содержать сведения о тех правилах, которые нельзя нарушать при работе с данными, и обладать инструментами контроля за тем, чтобы данные и их изменения соответствовали заданным правилам.

## 2. Критерии выбора и структура систем управления базами данных.

Выбор СУБД представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке приложений БД. Перечень требований к СУБД может изменяться в зависимости от поставленных целей. Тем не менее, можно выделить несколько групп критериев:

### 1. Моделирование данных.

1.1. Используемая модель данных. Существует множество моделей данных; самые распространенные – иерархическая, сетевая, реляционная,

объектно-реляционная и объектная. Вопрос об использовании той или иной модели должен решаться на начальном этапе проектирования БД.

1.2. Триггеры и хранимые процедуры. Триггер – программа БД, вызываемая при вставке, изменении или удалении строки таблицы. Триггеры обеспечивают проверку любых изменений на корректность, прежде чем эти изменения будут приняты. Хранимая процедура – программа, которая хранится на сервере и может вызываться клиентом. Поскольку хранимые процедуры выполняются непосредственно на сервере БД, обеспечивается более высокое быстродействие, нежели при выполнении тех же операций средствами клиента БД. В различных программных продуктах для реализации триггеров и хранимых процедур используются различные инструменты.

1.3. Средства поиска. Некоторые современные СУБД имеют встроенные дополнительные средства контекстного поиска.

1.4. Предусмотренные типы данных. Здесь следует учесть два фактически независимых критерия: базовые или основные типы данных, заложенные в систему, и наличие возможности расширения типов. В то время как отклонения базовых наборов типов данных у современных систем от некоего стандартного, обычно, невелики, механизмы расширения типов данных в системах того или иного производителя существенно различаются.

1.5. Реализация языка запросов. Все современные системы совместимы со стандартным языком доступа к данным SQL-92, однако многие из них реализуют те или иные расширения данного стандарта.

## 2. Особенности архитектуры и функциональные возможности.

2.1. Мобильность. Мобильность – это независимость системы от среды, в которой она работает. Средой в данном случае является как аппаратура, так и программное обеспечение (операционная система).

2.2. Масштабируемость. При выборе СУБД необходимо учитывать, сможет ли данная система соответствовать росту БД, причем рост может проявляться в увеличении числа пользователей, объема хранимых данных и объеме обрабатываемой информации.

2.3. Распределенность. Самый простой и надежный подход – централизация хранения и обработки данных на одном сервере. К сожалению, это не всегда возможно и приходится применять распределенные БД. Различные системы имеют разные возможности управления распределенными БД.

2.4. Сетевые возможности. Многие системы позволяют использовать широкий диапазон сетевых протоколов и служб для работы и

администрирования.

### 3. Контроль работы системы.

3.1. Контроль использования памяти компьютера. Система может иметь возможность управления использованием как оперативной памяти, так и дискового пространства. Во втором случае это может выражаться, например, в сжатии БД, удалении избыточных файлов.

3.2. Автонастройка. Многие современные системы включают в себя возможности самоконфигурирования, которые, как правило, опираются на результаты работы сервисов самодиагностики производительности. Данная возможность позволяет выявить слабые места конфигурации системы и автоматически настроить ее на максимальную производительность.

### 4. Особенности разработки приложений.

4.1. Многие производители СУБД выпускают также средства разработки приложений для своих систем. Как правило, эти средства позволяют наилучшим образом реализовать все возможности сервера, поэтому при анализе СУБД стоит рассмотреть также и возможности средств разработки приложений.

4.2. Средства проектирования. Некоторые системы имеют средства автоматического проектирования, как БД, так и прикладных программ. Средства проектирования различных производителей могут существенно различаться.

4.3. Многоязыковая поддержка. Поддержка большого количества национальных языков расширяет область применения системы и приложений, построенных на ее основе.

4.4. Возможности разработки Web-приложений. При разработке различных приложений зачастую возникает необходимость использовать возможности среды Internet. Средства разработки некоторых производителей имеют большой набор инструментов для построения приложений под Web.

4.5. Поддерживаемые языки программирования. Широкий спектр используемых языков программирования повышает доступность системы для разработчиков, а также может существенно повлиять на быстродействие и функциональность создаваемых приложений.

### 5. Производительность.

5.1. Рейтинг ТРС (Transactions per Cent). Для тестирования производительности применяются различные средства, и существует множество тестовых рейтингов. Одним из самых популярных и объективных является ТРС-анализ производительности систем. Фактически ТРС-анализ

рассматривает композицию СУБД и аппаратуры, на которой эта СУБД работает. Показатель ТРС – это отношение количества запросов обрабатываемых за некий промежуток времени к стоимости всей системы.

В составе СУБД выделяются программные, языковые, технические, организационно-методические и «человеческие» средства (рис. 1.5).



Рисунок 1.5. Состав СУБД

Программные средства включают трансляторы и систему управления, обеспечивающую ввод–вывод, обработку и хранение информации, создание, модификацию и тестирование БД.

Базовыми внутренними языками программирования являются языки четвертого поколения. В качестве базовых языков могут использоваться С, С++, Pascal, ObjectPascal. Язык С++ позволяет строить программы как на языке VisualBasic с его широким спектром возможностей, более близкий и понятный даже пользователю-непрофессионалу, так и на непроецедурном (декларативном) языке структурированных запросов SQL.

Исторически для СУБД сложились три языка: 1) язык описания данных (ЯОД), называемый также языком описания схем – для построения структуры («шапки») таблиц БД; 2) язык манипулирования данными (ЯМД) – для заполнения БД данными и операций обновления (запись, удаление, модификация); 3) язык запросов – язык поиска наборов величин в файле в соответствии с заданной совокупностью критериев поиска и выдачи затребованных данных без изменения содержимого файлов и БД (язык преобразования критериев в систему команд). В настоящее время функции всех трех языков выполняет язык SQL, относящийся к классу языков, базирующихся на исчислении кортежей. Вместе с тем сохранились и языки запросов, например, язык запросов по примеру QueryByExample (QBE) класса реляционного исчисления доменов. С помощью языков БД создаются

приложения, базы данных и интерфейс пользователя, включающий экранные формы, меню, отчеты. При создании БД на базе СУБД Paradox эти элементы (объекты) фиксируются в отдельных файлах. В СУБД Access, Interbase все созданные объекты размещаются в одном файле. Для работы с созданной БД пользователю или администратору БД следует иметь перечень файлов – таблиц с описанием состава их данных (структуры, схемы). Для этого создается специальный файл, называемый словарем данных (репозитарием, словарем-справочником, энциклопедией). Описание БД относится к метайнформации.

В качестве технических средств могут выступать супер- или персональные компьютеры с соответствующими периферийными устройствами. Организационно-методические средства – это совокупность инструкций, методических и регламентирующих материалов, описаний структуры и процедуры работы пользователя с СУБД и БД.

Компонентная структура СУБД очень различается в разных системах. Однако при изучении СУБД полезно представлять себе ее обобщенную структуру в виде набора нескольких компонент и определенных связей между ними.

СУБД состоит из нескольких программных компонентов (модулей), каждый из которых предназначен для выполнения специфической операции. Основные программные компоненты представлены на рисунке 1.6. На нем также показано, как СУБД взаимодействует с другими программными компонентами, например с такими, как пользовательские запросы и методы доступа (т. е. методы управления файлами, используемые при сохранении и извлечении записей с данными).

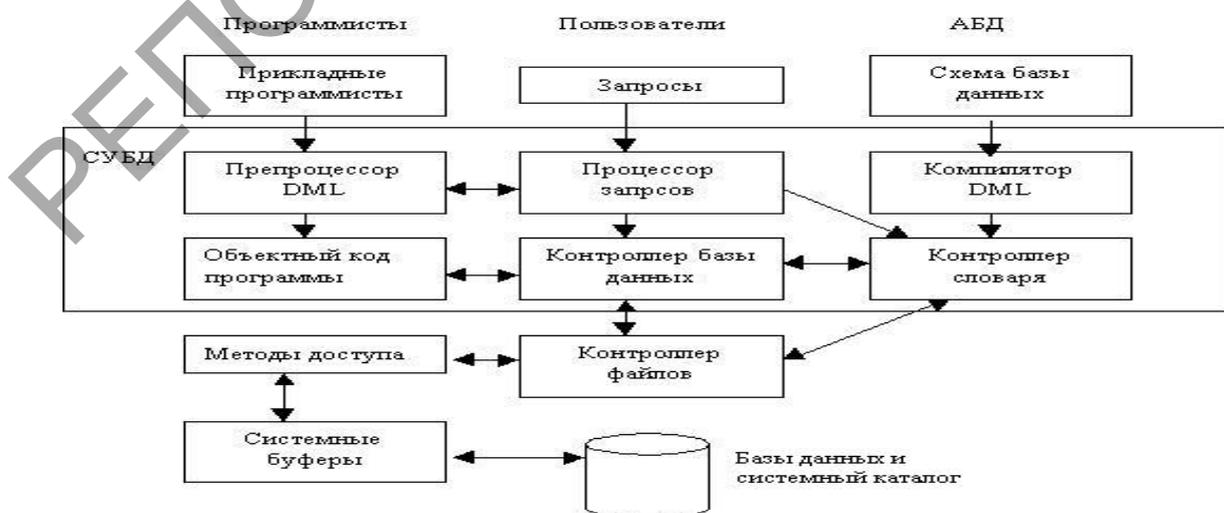


Рисунок 1.6. Основные программные компоненты типичной системы

*Процессор запросов.* Преобразует запросы в последовательность низкоуровневых инструкций для контроллера базы данных.

*Контроллер базы данных.* Этот компонент взаимодействует с запущенными пользователями прикладными программами и запросами. Контроллер БД принимает запросы и проверяет внешние и концептуальные схемы для определения тех концептуальных записей, которые необходимы для удовлетворения требований запроса. Затем контроллер БД вызывает контроллер файлов для выполнения поступившего запроса.

*Контроллер файлов.* Манипулирует предназначенными для хранения данных файлами и отвечает за распределение доступного дискового пространства. Он создает и поддерживает список индексов. Однако контроллер файлов не управляет физическим вводом и выводом данных непосредственно, а лишь передает запросы соответствующим методам доступа, которые считывают данные в системные буферы или записывают их на диск.

*Препроцессор языка DML.* Этот модуль преобразует внедренные в прикладные программы DML-операторы в вызовы стандартных функций базового языка.

*Компилятор языка DDL.* Компилятор языка DDL преобразует DDL-команды в набор таблиц, содержащих метаданные, затем эти таблицы сохраняются в системном каталоге.

*Контроллер словаря.* Контроллер словаря управляет доступом к системному каталогу и обеспечивает работу с ним.

Компонентную структуру контроллера базы данных, как основного модуля СУБД можно в обобщенном виде представить в виде следующей структурной схемы:

СУБД обладают как преимуществами по сравнению с файловыми системами, так и недостатками.

#### Преимущества СУБД:

1. Контроль за избыточностью данных. При использовании БД предпринимается попытка исключить избыточность данных за счет интеграции файлов, чтобы избежать хранения нескольких копий одного и того же элемента информации. Однако полностью избыточность информации в БД не исключается, а лишь контролируется ее степень.

2. Непротиворечивость данных. Если элемент данных хранится в БД только в одном экземпляре, то для изменения его значения требуется

выполнить только одну операцию обновления. Если элемент данных хранится в БД в нескольких экземплярах, то такая система сможет следить за тем, чтобы копии не противоречили друг другу.

3. Совместное использование данных. БД принадлежит группам пользователей или всей организации в целом и может совместно использоваться всеми зарегистрированными пользователями. При этом можно создавать новые приложения на основе уже существующей в БД информации и добавлять в нее только те данные, которые в настоящий момент еще не хранятся в ней, а не определять вновь требования ко всем данным, необходимым новому приложению.

4. Поддержка целостности данных. Целостность БД означает корректность и непротиворечивость хранимых в ней данных. Целостность обычно описывается с помощью ограничений, т.е. правил поддержки непротиворечивости, которые не должны нарушаться в БД. Ограничения можно применять к элементам данных внутри одной записи или к связям между записями.

5. Повышенная безопасность. Безопасность БД заключается в защите данных от несанкционированного доступа со стороны пользователей. Без привлечения соответствующих мер безопасности интегрированные данные становятся уязвимыми. СУБД приводит в действие систему безопасности БД. Система обеспечения безопасности может быть выражена в форме учетных имен и паролей для идентификации пользователей, которые зарегистрированы в БД.

6. Повышение эффективности с ростом масштабов системы. Комбинируя все рабочие данные в одной БД и создавая набор приложений, которые работают с одним источником данных, можно добиться существенной экономии средств.

7. Повышение доступности данных и их готовности к работе. Данные в результате интеграции становятся непосредственно доступными конечным пользователям. Во многих СУБД предусмотрены языки запросов или инструменты для создания отчетов, которые позволяют пользователям задавать непредусмотренные заранее вопросы и почти немедленно получать требуемую информацию на своих терминалах, не прибегая к помощи программиста.

8. Улучшение показателей производительности. На базовом уровне СУБД обеспечивает все низкоуровневые процедуры работы с файлами, которую обычно выполняют приложения. Во многих СУБД также предусмотрена среда разработки четвертого поколения с инструментами,

упрощающими создание приложений БД. Результатом является повышение производительности работы программистов и сокращение времени разработки новых приложений.

Недостатки СУБД:

1. Сложность. Обеспечение функциональности, которой должна обладать каждая СУБД, сопровождается ее значительным усложнением. Чтобы воспользоваться всеми преимуществами СУБД, проектировщики и разработчики БД, администраторы БД, а также конечные пользователи должны хорошо понимать функциональные возможности СУБД.

2. Размер программного обеспечения. Сложность и широта функциональных возможностей приводит к тому, что СУБД становится программным продуктом, который может занимать много места на диске и требовать большого объема оперативной памяти для эффективной работы.

3. Стоимость СУБД. В зависимости от имеющейся вычислительной среды и требуемых функциональных возможностей, стоимость СУБД может варьироваться в очень широких пределах – от нескольких сот до нескольких сот тысяч долларов. Кроме того, следует учесть ежегодные расходы на сопровождение системы, которые составляют некоторый процент от ее общей стоимости.

4. Дополнительные затраты на аппаратное обеспечение. Для удовлетворения требований, предъявляемых к дисковым накопителям со стороны СУБД и базы данных, может понадобиться приобрести дополнительные устройства хранения информации. Более того, для достижения требуемой производительности может понадобиться более мощный компьютер, который, возможно, будет работать только с СУБД.

5. Затраты на преобразование приложений. В некоторых ситуациях стоимость СУБД и дополнительного аппаратного обеспечения может оказаться несущественной по сравнению со стоимостью преобразования существующих приложений для работы с новой СУБД и новым аппаратным обеспечением. Эти затраты также включают стоимость подготовки персонала для работы с новой системой, а также оплату услуг специалистов, которые будут оказывать помощь в преобразовании и запуске новой системы.

6. Производительность. СУБД предназначены для решения общих задач и обслуживания нескольких приложений, а не одного из них. В результате многие приложения в новой среде будут работать не так быстро, как прежде.

7. Серьезные последствия при выходе системы из строя. Централизация ресурсов повышает уязвимость системы. Поскольку работа всех пользователей и приложений зависит от готовности к работе СУБД, выход из

стройка одного из ее компонентов может привести к полному прекращению всей работы СУБД.

### 3. Классификация систем управления базами данных.

СУБД классифицируют по различным признакам. Важнейшим классификационным признаком СУБД является *тип модели данных*, поддерживаемый СУБД. По этому признаку СУБД делятся на:

- иерархические;
- сетевые;
- реляционные;
- постреляционные;
- многомерные;
- объектно-ориентированные.

Иерархические СУБД поддерживают иерархическую модель БД, которая основана на древовидной структуре хранения информации. В этом смысле иерархические БД напоминают файловую систему компьютера.

В сетевой структуре каждый элемент может быть связан с любым другим элементом. Сетевые БД подобны иерархическим, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.

Реляционная СУБД – это СУБД, управляющая реляционными БД. В постреляционной модели СУБД сняты ограничения неделимости данных, хранящихся в записях таблиц, т.е. допускается наличие многозначных полей. В многомерной модели СУБД данные рассматриваются как кубы, которые являются обобщением электронных таблиц на любое число измерений.

Объектно-ориентированная СУБД реализует объектно-ориентированный подход. Эта система управления обрабатывает данные как абстрактные объекты, наделенные свойствами, в виде неструктурированных данных, и использующие методы взаимодействия с другими объектами окружающего мира. Объектно-ориентированные СУБД: Itaska (IBEX), Jasmine (Computer Associates), Matisse (ODB), Object Store (ODI), Ontos (Ontos), O2 (Ardent Software), Poet (Poet Software), Versant (Versant Technologies).

По архитектуре организации хранения данных СУБД делятся на:

- централизованные СУБД (все части СУБД размещаются на одном компьютере): dBase-подобные системы, DB2, Paradox, Access, FoxPro, Oracle, MSSQLServer;
- распределенные СУБД (части СУБД могут размещаться на двух и

более компьютерах): InformixOn-Line фирмы InformixSoftware; IngresIntelligentDatabase фирмы IngresCorp; Oracle(version 7) фирмы OracleCorp; SybaseSystem 10 фирмы SybaseInc.

СУБД классифицируются по способу доступа к БД на: файл-серверные, клиент-серверные и встраиваемые СУБД.

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. СУБД располагается на каждом клиентском компьютере (рабочей станции). Доступ СУБД к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок.

Преимуществом этой архитектуры является низкая нагрузка на процессор файлового сервера. Недостатки файл-серверных СУБД: потенциально высокая загрузка локальной сети; затруднённость или невозможность централизованного управления; затруднённость или невозможность обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность. Применяются чаще всего в системах с низкой интенсивностью обработки данных и низкими нагрузками на БД.

На данный момент файл-серверная технология считается устаревшей, а её использование в крупных информационных системах – недостатком.

Примеры файл-серверных СУБД: Microsoft Access, Paradox, dBase, FoxPro, Visual FoxPro.

Клиент-серверная СУБД позволяет обмениваться клиенту и серверу минимально необходимыми объёмами информации. При этом основная вычислительная нагрузка ложится на сервер. Клиент может выполнять функции предварительной обработки перед передачей информации серверу, но в основном его функции заключаются в организации доступа пользователя к серверу.

В большинстве случаев клиент-серверная СУБД гораздо менее требовательна к пропускной способности компьютерной сети, чем файл-серверная СУБД, особенно при выполнении операции поиска в базе данных по заданным пользователем параметрам, т.к. для поиска нет необходимости получать на клиент весь массив данных: клиент передаёт параметры запроса серверу, а сервер производит поиск по полученному запросу в локальной базе данных. Результат выполнения запроса, который обычно на несколько порядков меньше по объёму, чем весь массив данных, возвращается клиенту, который обеспечивает отображение результата пользователю.

Примеры клиент-серверных СУБД: Oracle, MySQL, MSSQLServer, IBMDB2, Informix, ЛИНТЕР.

Встраиваемые СУБД являются составной частью некоторого программного продукта и не требуют процедуры самостоятельной установки. Встраиваемая СУБД предназначена для локального хранения данных своего приложения и не рассчитана на коллективное использование в сети. Физически такая СУБД чаще всего реализована в виде подключаемой библиотеки. Доступ к данным со стороны приложения может происходить через SQL, либо через специальные программные интерфейсы.

Примерами встраиваемых СУБД являются OpenEdge, SQLite, BerkeleyDB, MicrosoftSQLServerCompactи др.

По *степени универсальности* различаются два класса СУБД – системы общего назначения и специализированные системы. СУБД общего назначения ориентированы на какую-либо конкретную предметную область или на информационные потребности конкретной группы пользователей. СУБД общего назначения обладает средствами настройки на работу с конкретной БД в условиях конкретного применения. В некоторых ситуациях СУБД общего назначения не позволяют добиться требуемых проектных и эксплуатационных характеристик (производительность, занимаемый объем памяти и прочее). Тем не менее, создание специализированных СУБД весьма трудоемкий процесс и для того, чтобы его реализовать, нужны очень веские основания.

По *языкам общения* СУБД делятся на открытые, замкнутые и смешанные. Открытые системы – это системы, в которых для обращения к БД используются универсальные языки программирования. Замкнутые системы имеют собственные языки общения с пользователями БД. Открытые системы в настоящее время используются редко.

По *выполняемым функциям* СУБД делятся на информационные и операционные. Информационные СУБД позволяют организовать хранение информации и доступ к ней. Операционные СУБД выполняют достаточно сложную обработку, например, автоматически позволяют получать агрегированные показатели, не хранящиеся непосредственно в БД, могут изменять алгоритмы обработки и т.д.

По *сфере возможного применения* различают универсальные и специализированные СУБД.

Системы управления базами данных поддерживают разные типы данных. *Набор типов данных*, допустимых в разных СУБД, различен. Ряд СУБД позволяет разработчику (прикладному программисту)

или администратору БД) добавлять новые типы данных и новые операции над этими данными. Такие системы называются расширяемыми системами баз данных (РСБД). Дальнейшим развитием концепции РСБД являются объектно-ориентированные системы баз данных, обладающие достаточно мощными возможностями, чтобы непосредственно моделировать сложные объекты.

По «мощности» СУБД делятся на «настольные» и «корпоративные». Характерными чертами «настольных» СУБД являются сравнительно невысокие требования к техническим средствам, ориентация на конечного пользователя, низкая стоимость. «Корпоративные» СУБД обеспечивают работу в распределенной среде, высокую производительность, поддержку коллективной работы при проектировании систем, имеют развитые средства администрирования и более широкие возможности поддержания целостности. В связи с этим очевидно, что корпоративные СУБД сложны, дороги, требуют значительных вычислительных ресурсов.

Наиболее известными из «корпоративных» СУБД являются Oracle, Informix, Sybase, MS SQL Server, Progress, DB2 и некоторые другие.

По ориентации на преобладающую категорию пользователей можно выделить СУБД для разработчиков и для конечных пользователей. Системы, относящиеся к первому классу, должны иметь качественные компиляторы и позволять создавать «отчуждаемые» программные продукты, обладать развитыми средствами отладки и обладать другими возможностями, позволяющими создавать эффективные сложные системы. Основными требованиями, предъявляемыми к системам, ориентированным на конечного пользователя, являются: удобство интерфейса, высокий уровень языковых средств, наличие интеллектуальных модулей подсказок, повышенная защита от непреднамеренных ошибок и т.п.

По характеру использования СУБД делят на персональные и многопользовательские. Персональные СУБД обычно обеспечивают возможность создания персональных БД и недорогих приложений, работающих с ними. К персональным СУБД, например, относятся Visual FoxPro, Paradox, Clipper, dBase, Access и др. Многопользовательские СУБД включают в себя сервер БД и клиентскую часть и, как правило, могут работать в неоднородной вычислительной среде (с разными типами компьютеров и разными операционными системами). К многопользовательским СУБД относятся, например, СУБД Oracle и Informix.

По своей архитектуре СУБД делятся на одно-, двух- и трехзвенные

(рис. 1.7). В однозвенной архитектуре используется единственное звено (клиент), обеспечивающее необходимую логику управления данными и их визуализацию. В двухзвенной архитектуре значительную часть логики управления данными реализует сервер баз данных (сервер БД), в то время как клиентское звено в основном занято отображением данных в удобном для пользователя виде. В трехзвенных СУБД используется промежуточное звено – сервер приложений. Сервер приложений позволяет полностью избавить клиента от функций по управлению данными и обеспечению связи с сервером БД.

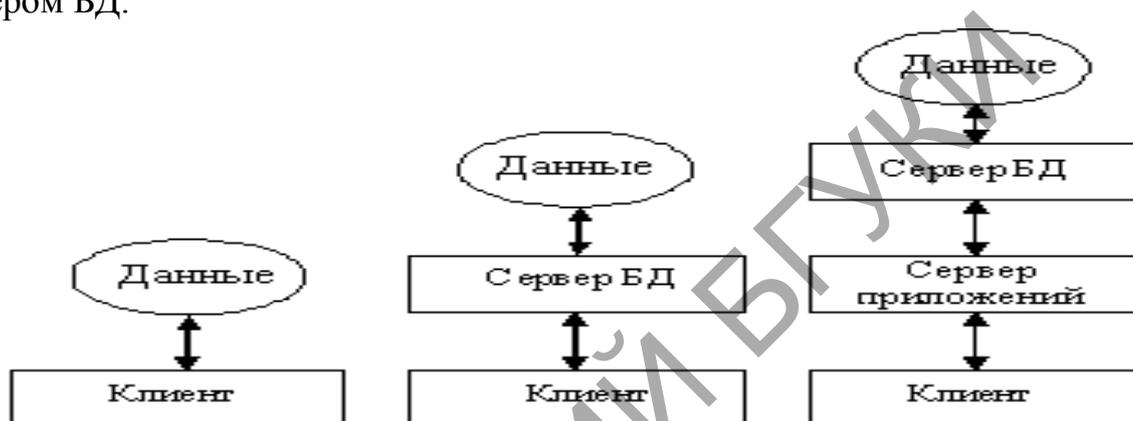


Рисунок 1.7. Классификация СУБД по архитектуре

В качестве классификационных признаков СУБД можно также рассматривать:

- среду функционирования СУБД (платформу) – класс компьютеров и операционных систем, под управлением которых работает СУБД;
- наличие диалоговых и инструментальных средств конструирования объектов БД;
- возможности встроенного языка СУБД;
- использование OLE-технологии – взаимодействие объектов БД с объектами других приложений: табличных и текстовых процессоров, графических редакторов и др.;
- обеспечение интеграции данных из баз, созданных в разных СУБД.

### Тема3. Архитектура систем управления базами данных

#### План:

1. Трехуровневая архитектура систем баз данных ANSI/SPARC.
2. Языки баз данных.

### 3. Архитектура многопользовательских систем управления базами данных.

#### 1. Трехуровневая архитектура систем баз данных ANSI/SPARC.

Архитектура систем БД была предложена подкомитетом SPARC (Standards Planning and Requirements Committee, комитет по планированию стандартов) американского национального института стандартов ANSI в 1975 году.

Под архитектурой системы БД понимается совокупность ее функциональных компонентов, а также средств обеспечения их взаимодействия друг с другом, с пользователями и с системным персоналом.

Архитектура систем БД ANSI/SPARC описывает логическую организацию системы с точки зрения представления данных пользователям. Одна и та же БД в зависимости от точки зрения может иметь различные уровни описания. По числу уровней описания данных, поддерживаемых СУБД, различают одно-, двух- и трехуровневые системы. В настоящее время чаще всего поддерживается трехуровневая архитектура описания БД (рис.

1.7), с тремя уровнями абстракции, на которых можно рассматривать БД.

Такая архитектура включает:

– *внешний уровень*, на котором пользователи воспринимают данные и отдельные группы пользователей имеют свое представление на БД;

– *внутренний уровень*, на котором СУБД и операционная система воспринимают данные;

– *концептуальный уровень* представления данных, предназначенный для отображения внешнего уровня на внутренний уровень, а также для обеспечения необходимой их независимости друг от друга; он связан с обобщенным представлением пользователей.

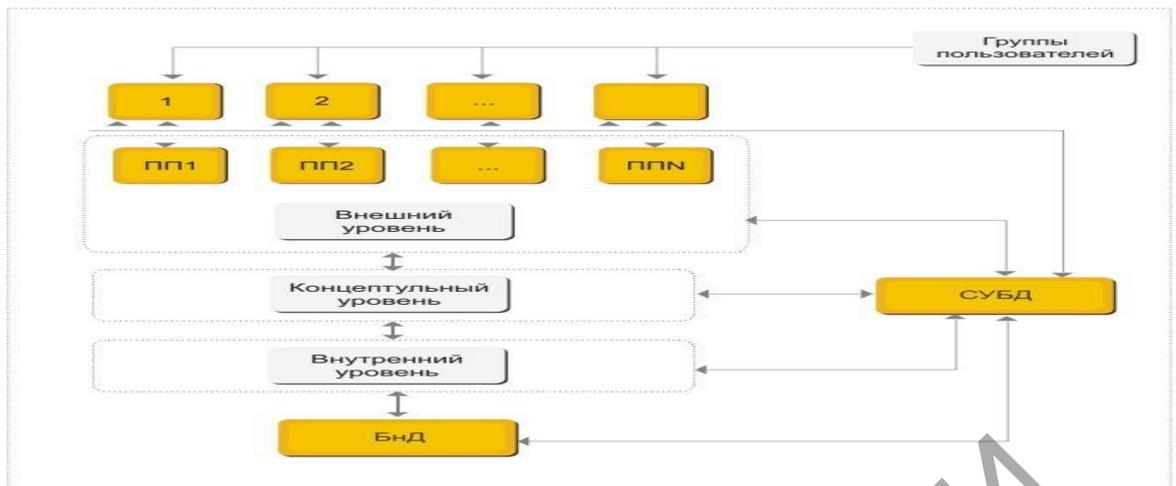


Рисунок 1.7. Трехуровневая архитектура СУБД

Описание структуры данных на любом уровне называется *схемой*. Существует три различных типа схем БД, которые определяются в соответствии с уровнями абстракции трехуровневой архитектуры. На самом высоком уровне имеется несколько внешних схем или подсхем, которые соответствуют разным представлениям данных. На концептуальном уровне описание БД называют *концептуальной схемой*, а на самом низком уровне абстракции — *внутренней схемой*.

Основным назначением трехуровневой архитектуры является обеспечение независимости от данных. Суть этой независимости заключается в том, что изменения на нижних уровнях никак не влияют на верхние уровни. Различают два типа независимости от данных: логическую и физическую.

*Логическая независимость от данных* означает полную защищенность внешних схем от изменений, вносимых в концептуальную схему. Такие изменения концептуальной схемы, как добавление или удаление новых сущностей, атрибутов или связей, должны осуществляться без необходимости внесения изменений в уже существующие внешние схемы для других групп пользователей.

*Физическая независимость от данных* означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему. Такие изменения внутренней схемы, как использование различных файловых систем или структур хранения, разных устройств хранения, модификация индексов или хеширование, должны осуществляться

без необходимости внесения изменений в концептуальную или внешнюю схемы.

Рассмотрим каждый из трех названных уровней.

*Внешний уровень* – это пользовательский уровень. Пользователем может быть программист, конечный пользователь или администратор БД. Представление БД с точки зрения пользователей называется внешним представлением. Внешнее представление – это содержимое БД, каким его видят определенный конечный пользователь или группа пользователей.

Внешних представлений обычно бывает несколько. Каждая группа пользователей выделяет в моделируемой предметной области, общей для всей организации, те сущности, атрибуты и связи, которые ей интересны. Например, пользователь из отдела кадров может рассматривать БД как набор записей об отделах и служащих. Он может ничего не знать про записи о деталях и поставщиках, с которыми работают пользователи в отделе обеспечения. Эти частичные описания БД для отдельных групп пользователей или ориентированные на отдельные аспекты предметной области называются подсхемой.

*Концептуальный уровень* является промежуточным уровнем в трехуровневой архитектуре и обеспечивает представление всей информации БД в абстрактной форме. Концептуальное представление – это представление всего содержимого БД. Описание БД на этом уровне называется концептуальной схемой, которая является результатом концептуального проектирования.

Концептуальная схема – это единое логическое описание всех элементов данных и отношений между ними, логическая структура всей БД. Для каждой БД имеется только одна концептуальная схема.

Концептуальная схема должна содержать:

- сущности и их атрибуты;
- связи между сущностями;
- ограничения, накладываемые на данные;
- семантическую информацию о данных;
- обеспечение безопасности и поддержки целостности данных.

*Внутренний уровень* является третьим уровнем архитектуры БД. Внутреннее представление описывает все подробности, связанные с хранением данных в базе. Внутреннее представление описывается с помощью внутренней схемы. Внутренняя схема раскрывает физическую реализацию БД и предназначена для достижения оптимальной

производительности и обеспечения экономного использования дискового пространства. Для каждой БД существует только одна внутренняя схема.

На внутреннем уровне хранится следующая информация:

- распределение дискового пространства для хранения данных и индексов;
- описание подробностей сохранения записей (с указанием реальных размеров сохраняемых элементов данных);
- сведения о размещении записей;
- сведения о сжатии данных и выбранных методах их шифрования.

СУБД отвечает за установление соответствия между всеми тремя типами схем разных уровней, а также за проверку их непротиворечивости.

Ниже внутреннего уровня находится *физический уровень*, который контролируется операционной системой, но под руководством СУБД. Физический уровень учитывает, каким образом данные будут представлены в компьютере. Он обеспечивает физический взгляд на БД: дисководы, физические адреса, индексы, указатели и т.д. За этот уровень отвечают проектировщики физической БД, которые работают только с известными операционной системе элементами. Однако функции СУБД и операционной системы на физическом уровне не вполне четко разделены и могут варьироваться от системы к системе.

Архитектура ANSI/SPARC имеет большое теоретическое значение, определяя пути обеспечения логической и физической независимости данных.

## 2. Языки баз данных.

Язык для работы с данными, который способна воспринимать СУБД, состоит из двух частей: *языка определения данных* (Data Definition Language – DDL) и *языка манипулирования данными* (Data Manipulation Language – DML). Эти языки называются *подъязыками данных*, поскольку в них отсутствуют конструкции для выполнения вычислительных операций, обычно используемых в языках программирования высокого уровня: условные операторы или операторы цикла.

Язык определения данных (ЯОД) – описательный язык, с помощью которого описывается предметная область: именуются объекты, определяются их свойства и связи между объектами. Он используется, главным образом, для определения логической структуры БД. Схема базы данных, выраженная в терминах языка определения данных, состоит из

набора определений. ЯОД используется как для определения новой схемы, так и для модификации уже существующей. Результатом компиляции ЯОД-операторов является набор таблиц, хранимый в системном каталоге, в котором содержатся метаданные – т.е. данные, которые включают определения записей, элементов данных, а также другие объекты, представляющие интерес для пользователей или необходимые для работы СУБД. Перед доступом к реальным данным СУБД обычно обращается к системному каталогу.Синонимами термина «системный каталог» являются термины «словарь данных» и «каталог данных».

Язык манипулирования данными (ЯМД) содержит набор операторов манипулирования данными, т. е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные. Множество операций над данными можно классифицировать следующим образом:

1. операции селекции;
2. действия над данными:
  - включение – ввод экземпляра записи в БД с установкой его связей;
  - удаление – исключение экземпляра записи из БД с установкой новых связей; □
  - модификация – изменение содержимого экземпляра записи и коррекция связей при необходимости.

Таким образом, одна из основных функций СУБД заключается в поддержке ЯМД, с помощью которого пользователь может создавать выражения для выполнения перечисленных выше операций с данными.

ЯМД делятся на два типа. Это разделение обусловлено коренным различием в подходах к работе с данными, а следовательно, различием в базовых конструкциях в работе с данными.

Первый тип – это *процедурный ЯМД*.

Второй тип — это *декларативный (непроцедурный) ЯМД*.

Основное различие между этими типами языков заключается в том, что процедурные языки указывают то, *как* можно получить необходимый результат, тогда как непроцедурные языки описывают то, *какой* результат требуется получить. Как правило, в процедурных языках записи рассматриваются по отдельности, тогда как непроцедурные языки оперируют с целыми наборами записей.

*Процедурный ЯМД* сообщает системе о том, какие данные необходимы, и точно указывает, как их можно извлечь. С помощью процедурного ЯМД пользователь, а точнее – программист, указывает, какие данные ему

необходимы и как их можно получить. Это значит, что пользователь должен определить все операции доступа к данным (осуществляемые посредством вызова соответствующих процедур), которые должны быть выполнены для получения требуемой информации. Обычно такой процедурный ЯМД позволяет извлечь запись, обработать ее и, в зависимости от полученных результатов, извлечь другую запись, которая должна быть подвергнута аналогичной обработке, и т.д. Подобный процесс извлечения данных продолжается до тех пор, пока не будут извлечены все запрашиваемые данные. ЯМД сетевых и иерархических СУБД обычно являются процедурными.

К процедурным ЯМД относятся и языки, поддерживающие операции реляционной алгебры, которую основоположник теории реляционных баз данных Э.Ф. Кодд ввел для управления реляционной БД. Реляционная алгебра – это процедурный язык обработки реляционных таблиц, где в качестве операндов выступают таблицы в целом.

*Непроцедурный ЯМД* позволяет указать лишь то, какие данные требуются, но не то, как их следует извлекать. Непроцедурные ЯМД задают весь набор требуемых данных с помощью одного оператора выборки или обновления. С помощью непроцедурных языков DML пользователь указывает, какие данные ему нужны, без определения способа их получения. СУБД транслирует выражение на языке DML в процедуру (или набор процедур), которая обеспечивает манипулирование указанным набором записей. Такой подход освобождает пользователя от необходимости знать подробности внутренней реализации структур данных и особенности алгоритмов, используемых для извлечения и возможного преобразования данных. В результате работа пользователя становится в определенной степени независимой от данных. Непроцедурные языки часто также называют *декларативными языками*. Реляционные СУБД в той или иной форме обеспечивают поддержку непроцедурных языков манипулирования данными. Непроцедурные языки обычно проще понять и использовать, чем процедурные языки, поскольку пользователю достается меньшая часть работы, а СУБД – большая.

В настоящее время нормой является поддержка декларативного языка SQL, в основе которого лежит реляционное исчисление, введенное Э. Коддом. Этот язык стал стандартом для языков реляционных БД, что позволяет использовать один и тот же синтаксис и структуру команд при переходе от одной СУБД к другой. Язык SQL имеет сразу два компонента: язык DDL (ЯОД) и язык DML (ЯМД). Другим широко используемым языком

обработки данных является язык QBE, который заслужил репутацию одного из самых простых способов извлечения информации из БД. Особенно это ценно для пользователей, не являющихся профессионалами в этой области. Язык предоставляет графические средства создания запросов на выборку данных с использованием шаблонов. Ответ на запрос также представляет собой графическую информацию.

Часть непроцедурного языка ЯМД, которая отвечает за извлечение данных, называется языком запросов. *Язык запросов* можно определить как высокоуровневый узкоспециализированный язык, предназначенный для удовлетворения различных требований по выборке информации из базы данных

Выделяются *языки 4GL*. Аббревиатура «4GL» представляет собой сокращенный английский вариант написания термина «язык четвертого поколения» (Fourth-Generation Language). Если для организации некоторой операции с данными на языке третьего поколения (3GL) типа C++ потребуется написать сотни строк кода, то для реализации этой же операции на языке четвертого поколения достаточно 10–20 строк. В то время как языки третьего поколения являются процедурными, языки 4GL выступают как непроцедурные, поскольку пользователь определяет, что должно быть сделано, но не сообщает, как именно должен быть получен желаемый результат. Реализация языков четвертого поколения в значительной мере основана на использовании компонентов высокого уровня. Пользователю не требуется определять все этапы выполнения программы, необходимые для решения поставленной задачи, а достаточно лишь задать нужные параметры, на основании которых упомянутые выше компоненты автоматически осуществят генерацию приложения. В ряде случаев языки четвертого поколения позволяют повысить производительность труда на порядок, но за счет ограничения типов задач, которые можно будет решать с их помощью. Выделяют следующие типы языков четвертого поколения:

- языки представления информации, например языки запросов или генераторы отчетов;
- специализированные языки, например языки электронных таблиц и БД;
- генераторы приложений, которые при создании приложений обеспечивают определение, вставку, обновление или извлечение сведений из БД;
- языки очень высокого уровня, предназначенные для генерации кода приложений.

В качестве примеров языков четвертого поколения для работы с БД можно указать упомянутые выше языки *SQL* и *QBE*, а также генераторы форм, генераторы отчетов, генераторы графического представления данных, генераторы приложений.

*Генератор форм* представляет собой интерактивный инструмент, предназначенный для быстрого создания шаблонов ввода и отображения данных в экранных формах. Генератор форм позволяет пользователю определить внешний вид экранной формы, ее содержимое и место расположения на экране. С его помощью можно задавать цвета элементов экрана, а также другие характеристики.

*Генератор отчетов* является инструментом создания отчетов на основе хранимой в БД информации. Он подобен языку запросов в том смысле, что пользователю предоставляются средства создания запросов к БД и извлечения из нее информации, используемой для представления в отчете. Однако генераторы отчетов, как правило, предусматривают большие возможности управления внешним видом отчета. Генератор отчета позволяет либо автоматически определять вид получаемых результатов, либо с помощью специальных команд создавать свой собственный вариант внешнего вида документа.

Существуют два основных типа генераторов отчетов: языковой и визуальный. В первом случае для определения нужных для отчета данных и внешнего вида документа следует ввести соответствующую команду на некотором подязыке. Во втором случае для этих целей используется визуальный инструмент, подобный генератору форм.

*Генератор графического представления данных* – инструмент, предназначенный для извлечения информации из БД и отображения ее в виде диаграмм с графическим представлением существующих тенденций и связей. Обычно с помощью подобного генератора создаются гистограммы, круговые, столбчатые, точечные диаграммы и т.д.

*Генератор приложений* представляет собой инструмент для создания программ, взаимодействующих с БД. Применяя генератор приложений, можно сократить время, необходимое для проектирования полного объема требуемого прикладного программного обеспечения. Генераторы приложений обычно состоят из предварительно созданных модулей, содержащих фундаментальные функции, которые требуются для работы большинства программ. Эти модули, обычно создаваемые на языках высокого уровня, образуют «библиотеку» доступных функций. Пользователь

указывает, какие задачи программа должна выполнить, а генератор приложений определяет, как их следует выполнить в генерируемой программе. Примером генератора приложений является мастер форм БД, имеющийся в системе визуального программирования Delphi.

### 3. Архитектура многопользовательских систем управления базами данных.

В несколько абстрактной форме можно представить многопользовательскую систему БД совокупностью двух компонент – создающего запросы *клиента*, и *сервера*, который выполняет входящие запросы. В частном случае, и клиентское, и серверное приложения могут выполняться на одном и том же компьютере. Но они могут находиться и на разных компьютерах, связанных телекоммуникационной сетью. При этом предполагается, что один сервер может обслужить множество клиентов (рис. 1.8).

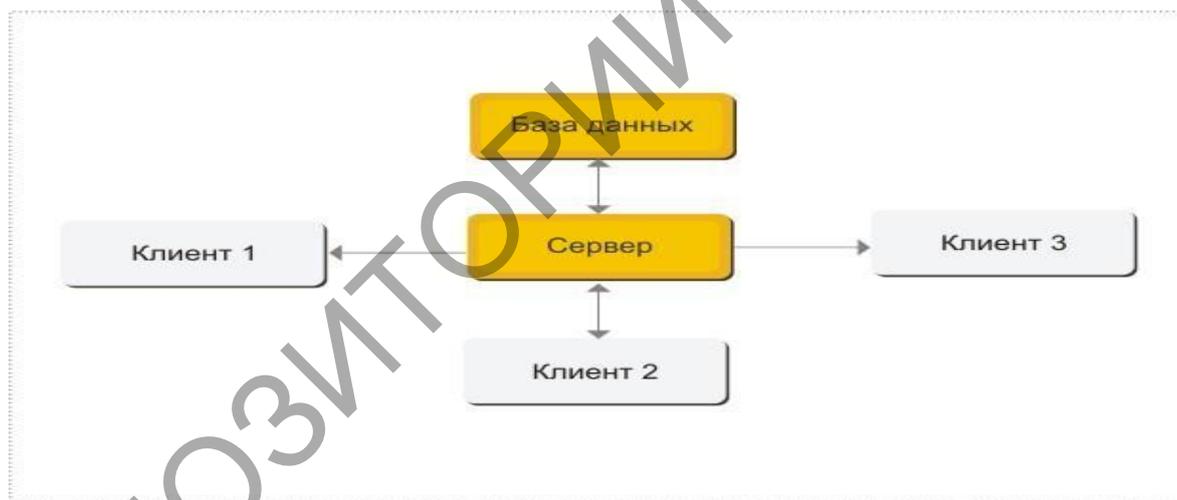


Рисунок 1.8. Структура системы БД с выделением клиентов и сервера

*Сервер* – это сама СУБД. Он поддерживает все основные функции СУБД, а именно: определение данных, манипулирование данными, защиту данных, поддержание целостности данных и т.д. В частности, он предоставляет полную поддержку внешнего, концептуального и внутреннего уровней. Поэтому *сервер* в этом контексте – это просто другое название для СУБД.

*Клиенты* – это различные приложения, которые выполняются с помощью СУБД. Таковыми являются как приложения, написанные пользователями, так и встроенные приложения, предоставляемые поставщиками СУБД или некоторыми сторонними

поставщиками программного обеспечения.

Обычно в приложении выделяются следующие группы функций:

- функции ввода и отображения данных;
- прикладные функции, определяющие основные алгоритмы решения задач приложения;
- функции обработки данных внутри приложения;
- функции управления информационными ресурсами;
- служебные функции, играющие роль связок между функциями первых четырех групп.

В зависимости от разделения функций между клиентом и сервером выделяются несколько типов архитектур систем БД.

*Файл-серверная архитектура.* Некоторые СУБД могут работать только в режиме «файл-сервер». Это означает, что СУБД находится на клиентской рабочей станции и даже может быть скомпонована с прикладным программным обеспечением. Для осуществления совместного доступа к данным файлы БД в этом случае размещаются на файловом сервере (рис. 1.9).

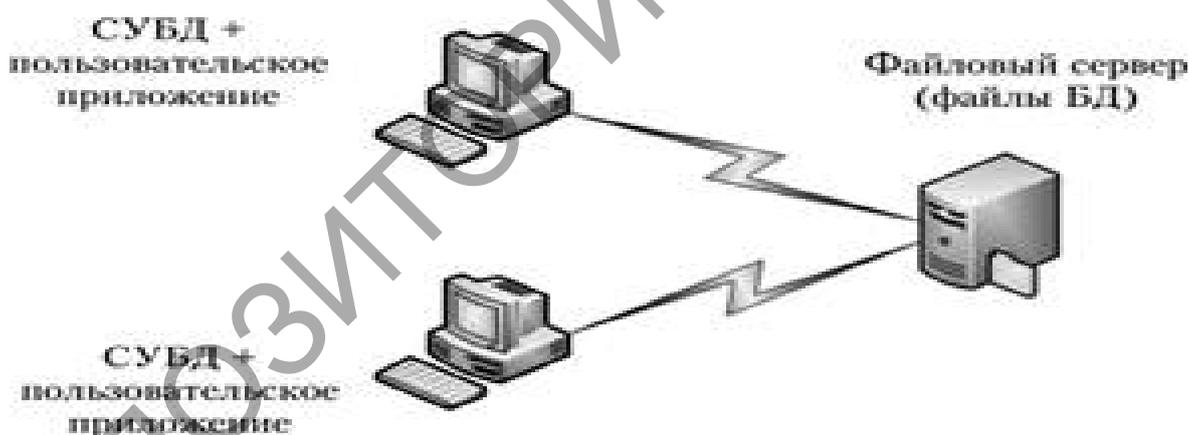


Рисунок 1.9. Файл-серверная архитектура

Если работа с данными ведется на нескольких компьютерах, то на каждом из них будет запущен экземпляр СУБД. Для обработки задания пользователя с сервера запрашиваются файлы с данными, а их обработка производится локально. В результате в таких системах приходится передавать по сети много данных для того, чтобы на стороне клиента среди них найти те, которые удовлетворяют запросу. Такой подход характеризуется значительным сетевым трафиком, что может привести к снижению производительности всей системы в целом.

Помимо этого недостатка у файл-серверной архитектуры есть и другие:

- на каждой рабочей станции должна находиться полная копия СУБД;
- управление параллельностью, восстановлением и целостностью усложняется, поскольку доступ к одним и тем же файлам могут осуществлять сразу несколько экземпляров СУБД;
  - узкий спектр операций манипулирования данными, который определяется только файловыми командами;
  - защита данных осуществляется только на уровне файловой системы.

Если один из клиентов заблокировал часть записей (СУБД внесла информацию в файл блокировок, что эти записи будут изменяться и их нельзя использовать другим клиентам), после чего потерял связь с сервером, экземпляры СУБД на других клиентах не смогут использовать соответствующие фрагменты БД, пока эту проблему не решит администратор, откорректировав файл блокировок. Файл-серверная архитектура предъявляет дополнительные требования к производительности клиентских рабочих мест: вся обработка данных производится на клиенте.

Основное достоинство этой модели, заключается в том, что в ней уже осуществлено разделение монопольного приложения на два взаимодействующих процесса. При этом сервер может обслуживать множество клиентов, обращающихся к нему с запросами.

Примером многопользовательской системы с файл-серверной архитектурой является совместное использование БД Microsoft Access в локальной сети. В таком режиме с одной базой могут нормально работать несколько пользователей: в зависимости от размеров базы, характеристик сети и интенсивности работы с данными, это может быть до 5–10 одновременных сеансов.

*Двухзвенная архитектура «клиент–сервер».* Данная архитектура предполагает, что СУБД находится на сервере, и только она имеет доступ к файлам БД (рис. 1.10).

На клиентских компьютерах работают пользовательские приложения и клиентские компоненты СУБД, осуществляющие взаимодействие с сервером. От клиента на сервер приходят запросы, которые обрабатываются СУБД, и результат отправляется на клиентский компьютер. По сравнению с файл-серверной архитектурой, в этом случае минимизируется сетевой трафик: по сети передаются только затребованные данные. Централизованная работа с файлами БД позволяет более эффективно решать вопросы, связанные с совместным доступом к данным и с обеспечением безопасности. В связи с тем, что СУБД работает на сервере, снижаются и требования к оборудованию на стороне клиента. Прикладная программа может быть написана на любом

языке, который поддерживает взаимодействие с используемой СУБД через имеющиеся клиентские компоненты: могут использоваться такие технологии, как ODBC, ADO или ADO.Net, JDBC и др.

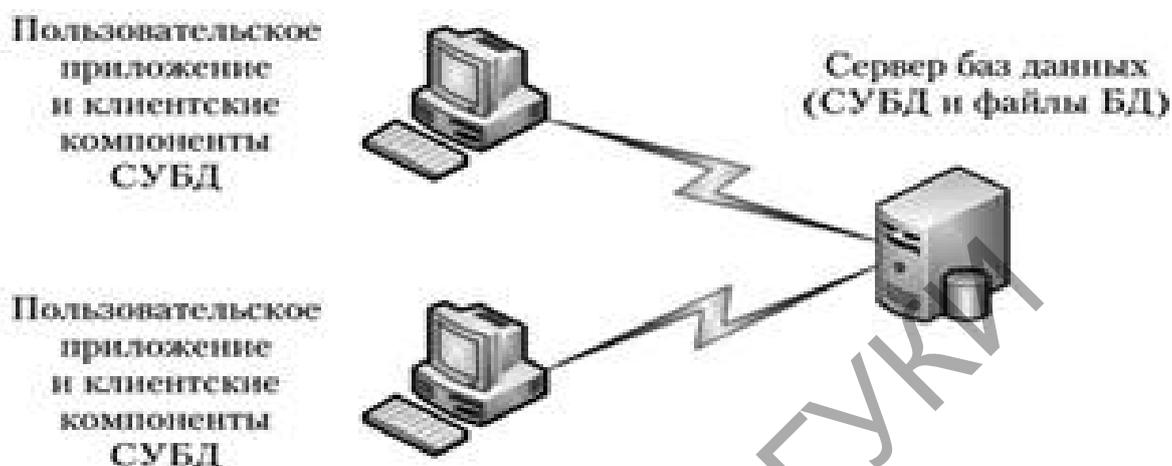


Рис. 1.10. Двухзвенная архитектура «клиент-сервер»

В двухзвенной архитектуре «клиент-сервер» функции клиента следующие:

- управляет пользовательским интерфейсом;
- принимает и проверяет синтаксис введенного пользователем запроса;
- выполняет приложение;
- генерирует запрос к БД и передает его серверу;
- отображает полученные данные пользователю.

Функции сервера:

- принимает и обрабатывает запросы к БД со стороны клиентов;
- проверяет полномочия пользователей;
- гарантирует соблюдение ограничений целостности;
- выполняет запросы обновления и возвращает результаты клиенту;
- поддерживает системный каталог;
- обеспечивает параллельный доступ к БД;
- обеспечивает управление восстановлением.

Двухзвенная архитектура широко используется для создания информационных систем, но и у нее есть ряд недостатков. Во-первых, при большом количестве и географической удаленности клиентских рабочих мест возникают проблемы с обновлением и устранением ошибок. Вторая проблема – каждый клиент независимо от других может в произвольный момент открыть соединение с СУБД. Сервер поддерживает открытые соединения со всеми активными клиентами, даже если никакой

работы нет. При большом числе клиентов это может негативно влиять на производительность сервера БД. Сервер играет пассивную роль, поэтому функции управления информационными ресурсами должны выполняться на клиенте.

*Трехзвенная архитектура.* Исправить недостатки двухзвенной архитектуры позволяет трехзвенная, также называемая трехуровневой (рис. 1.11). Данная архитектура предполагает наличие дополнительного сервера приложений, который проводит предварительную обработку запросов клиентов, формирует запросы к серверу БД и обрабатывает полученные результаты перед отправкой их клиенту.

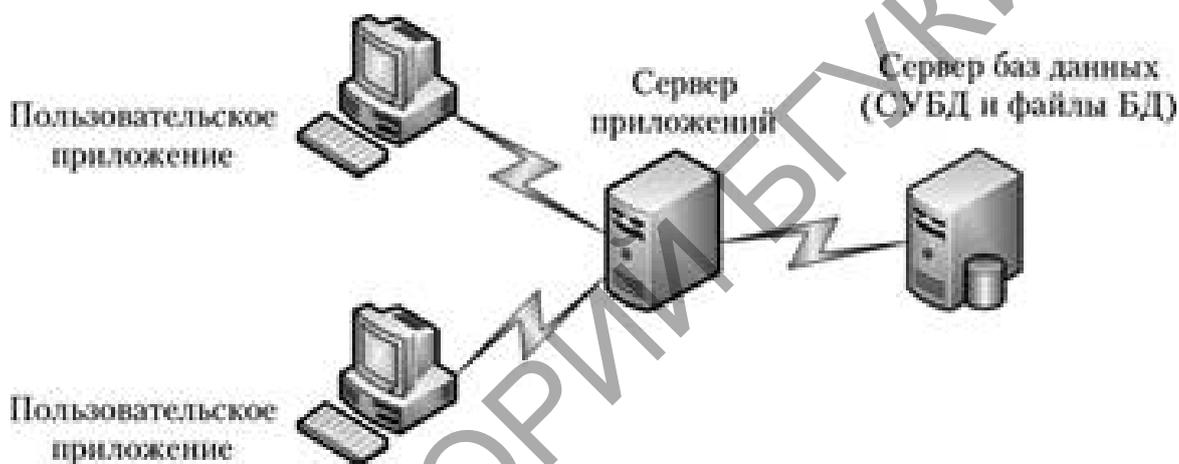


Рис. 1.11. Трехзвенная архитектура

В трехзвенной архитектуре большая часть логики приложения перенесена с клиента на сервер, и задачи клиентского приложения сводятся, в основном, к реализации пользовательского интерфейса и представлению результатов. В этой архитектуре только сервер приложений может подключаться к серверу БД. Это снимает проблему поддержания неиспользуемых соединений и более предпочтительно с точки зрения безопасности. Недостатком многоуровневой архитектуры «клиент-сервер» является сложность разработки подобных решений.

*Архитектура Интернет/Интранет-приложений.* В случае архитектуры интернет/интранет-приложений обязательным компонентом серверной части является веб-сервер, на котором выполняется веб-приложение, обращающееся к СУБД для доступа к БД (рис. 1.12). В роли универсального клиентского приложения выступает Интернет-браузер на устройстве клиента (в данном случае это может быть персональный компьютер, планшетный компьютер или какое-то мобильное устройство).



Рис. 1.12. Архитектура Интернет/Инtranет-решений

В небольших решениях веб-сервер может находиться на одном физическом компьютере с СУБД. В крупных системах, испытывающих большие нагрузки, задействуется множество веб-серверов и серверов БД.

Достоинства и недостатки подобной архитектуры связаны с использованием браузера в качестве универсального клиента. С одной стороны, отпадает необходимость в разработке, распространении и поддержке клиентского приложения. Появляется универсальность и частичная независимость от клиентской платформы. С другой стороны, в случае ограниченного числа пользователей приложения с помощью специально разработанного клиентского программного обеспечения можно добиться большей функциональности по сравнению с использованием веб-технологий.

#### Тема4. Модели представления данных

##### План:

1. Понятие «модель данных». Классификация моделей данных.
2. Иерархическая модель данных: основные понятия, форма представления, достоинства и недостатки.
3. Сетевая модель данных: основные понятия, форма представления, достоинства и недостатки.
4. Реляционная модель данных: основные понятия, форма представления, достоинства и недостатки.

1. Понятие «модель данных». Классификация моделей данных.

При работе с БД основными понятиями являются «данные» и «модель данных». Данные – это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы. Например, данными являются: Петров Николай Степанович, \$30 и т.д. Данные не обладают определенной структурой и становятся информацией тогда, когда пользователь задает им конкретную структуру, то есть вкладывает в них смысловое содержание. Поэтому центральным понятием в области БД является понятие модели. Не существует однозначного определения этого термина, у разных авторов эта абстракция определяется с некоторыми различиями, но, тем не менее, можно выделить нечто общее в этих определениях и определить модель данных как некоторую абстракцию, которая применительно к данным позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

С учетом трехуровневой архитектуры СУБД (см. рис. 1.7) понятие модели данных трансформируется для каждого уровня. Например, физическая модель данных оперирует категориями, касающимися организации внешней памяти и структур хранения, используемых в данной операционной среде. В качестве физических моделей применяются различные методы размещения данных, основанные на файловых структурах и страничной организации данных.

Первостепенную важность имеют модели данных, используемые на концептуальном уровне. По отношению к ним внешние модели называются подсхемами и используют те же абстрактные категории, что и концептуальные модели данных.

Классификация моделей данных отражена на рис. 1.13.



Рис. 1.13. Классификация моделей данных

При проектировании БД модель должна выразить информацию о предметной области в виде, независимом от используемой СУБД. Такая модель называется *инфологической* и отражает в естественной и удобной для разработчиков и других пользователей форме информационно-логический уровень абстрагирования, связанный с фиксацией и описанием объектов предметной области, их свойств и их взаимосвязей. *Инфологические модели* данных используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения, а *даталогические модели* уже поддерживаются конкретной СУБД.

*Документальные модели* данных соответствуют представлению о слабоструктурированной информации, ориентированной в основном на свободные форматы документов, текстов на естественном языке. Модели, ориентированные на формат документа, связаны, прежде всего, со стандартным общим языком разметки – SGML, который был утвержден ISO в качестве стандарта еще в 1980-е гг. Этот язык предназначен для создания других языков разметки, он определяет допустимый набор дескрипторов, их атрибуты и внутреннюю структуру документа. С помощью SGML можно описывать структурированные данные, организовывать информацию, содержащуюся в документах, представлять эту информацию в некотором стандартизованном формате.

Гораздо более простой и удобный, чем SGML, язык HTML позволяет определять оформление элементов документа и вносить специальные дескрипторы в документы, при помощи которых осуществляется процесс разметки. Дескрипторы на языке HTML, в первую очередь, предназначены для управления процессом вывода содержимого документа на экране с помощью программы-клиента (например, браузера) и определяют этим самым способ представления документа, но не его структуру. На языке HTML документ представляется набором элементов, причем начало каждого элемента, а в большинстве случаев и его конец, отмечается дескриптором, который называется тегом. В начале элемента указывается открывающий тег, а в конце – закрывающий. Например, элемент, соответствующий размечаемому документу, открывается тегом <html>, закрывается тегом </html> и содержит внутри себя элементы заголовка и тела документа, ограниченные специальными тегами <head></head> и <body></body>:

```
<html>
<head>
заголовок документа
</head>
<body>
    тело документа
</body>
</html>
```

В качестве компонента гипертекстовой базы данных, описываемой на языке HTML, используется текстовый файл, который может легко передаваться по сети с использованием протокола HTTP. Эта особенность, а также то, что HTML является открытым стандартом и огромное количество пользователей имеют возможность применять возможности этого языка для оформления документов, повлияли на рост популярности HTML и сделали его главным средством представления информации в Интернете.

Однако HTML сегодня уже не удовлетворяет в полной мере требованиям, предъявляемым современными разработчиками к языкам подобного рода. На смену ему пришел новый язык гипертекстовой разметки – язык XML.

XML – это расширяемый язык разметки, описывающий целый класс объектов данных, называемых XML-документами. Он используется в качестве средства для описания грамматики других языков и проверки правильности составления документов. Сам по себе XML не содержит никаких тегов, предназначенных для разметки, но определяет порядок их

создания.

*Тезаурусные модели* основаны на принципе организации словарей, содержат определенные языковые конструкции и принципы их взаимодействия в заданной грамматике. Эти модели эффективно используются в системах-переводчиках, особенно многоязыковых переводчиках. Принцип хранения информации в этих системах и подчиняется тезаурусным моделям.

*Дескрипторные модели* – самые простые из документальных моделей, они широко использовались на ранних стадиях использования документальных БД. В этих моделях каждому документу соответствовал дескриптор – описатель. Этот дескриптор имел жесткую структуру и описывал документ в соответствии с характеристиками, требуемыми для работы с документами в документальной БД.

К *теоретико-графовым моделям* относятся две разновидности:

- *сетевые модели*;
- *иерархические модели*.

В таких моделях данных предусматриваются характерные для подобного рода структур операции навигации и манипулирования данными.

Аппарат навигации в теоретико-графовых моделях служит для установки тех объектов данных, к которым будет применяться очередная операция манипулирования данными.

Теоретико-множественные модели используют математический аппарат, реляционную алгебру (знаковая обработка множеств), реляционное исчисление. К моделям данного типа относятся *реляционные модели*.

*Иерархическая, сетевая и реляционная модели* отражают способ установления связей между данными. Иерархическая и сетевая модели предполагают наличие связей между данными, имеющими какой-либо общий признак.

Использование иерархических и сетевых моделей ускоряет доступ к информации в БД. Но поскольку каждый элемент данных должен содержать ссылки на некоторые другие элементы, требуются значительные ресурсы как дисковой, так и основной памяти компьютера. Кроме того, для этих моделей характерна сложность реализации СУБД.

*Реляционная модель* является простейшей и наиболее привычной формой представления данных в виде таблицы. В теории множеств таблице соответствует термин «отношение» (relation), который и дал название реляционной модели. Достоинством реляционной модели является сравнительная простота инструментальных средств ее поддержки, а

недостатком – жесткость структуры данных и зависимость скорости выполнения операций от размера таблиц.

При создании моделей данных используются такие понятия, как «сущности», «атрибуты» и «связи». Сущность – это отдельный класс объектов предметной области (сотрудники или клиенты, понятия или события), который должен быть представлен в базе данных. Атрибут – это свойство, описывающее определенный аспект объекта, значение которого следует зафиксировать в описании предметной области. Связь является ассоциативным отношением между сущностями, при котором каждый экземпляр одной сущности соединен с некоторым (в том числе нулевым) количеством экземпляров другой сущности.

*Объектно-ориентированная модель* расширяет определение сущности с целью включения в него не только атрибутов, которые описывают состояние объекта, но и действий, которые с ним связаны, т.е. его поведение. В таком случае говорят, что объект инкапсулирует состояние и поведение.

2. Иерархическая модель данных: основные понятия, форма представления, достоинства и недостатки.

Первые иерархические и сетевые СУБД были созданы в начале 1960-х годов. Причиной послужила необходимость управления миллионами записей (связанных друг с другом иерархическим образом), например при информационной поддержке лунного проекта Аполлон. Из иерархических СУБД, самое большое распространение получила СУБД IMS (Information Management System компании IBM).

*Иерархическая модель данных* – модель с отношением подчиненности между данными. Отношение объектов иерархической модели определяется следующими правилами:

– (А подчинено Б) => (Б не подчинено А);

– (А подчинено Б) & (А подчинено В) => (Б тождественно В),

где А, Б, В – объекты. Другими словами, иерархическая модель реализуется древовидной структурой.

Объекты иерархических моделей представлены узлами деревьев, которыми описываются модели. Они представляют собой *записи*, которые также называются *сегментами*.

*Сегмент* или *запись* составляет основную единицу обработки БД: записи запоминаются, извлекаются, удаляются. Определяют *тип* и *экземпляр записи (сегмента)*. *Тип записи* – это именованная совокупность

полей данных с указанием их типов, *экземпляр записи* (или просто запись) – некоторая совокупность значений элементов в последовательности, соответствующей определению типа. Иными словами, *типзаписи* задает все множество подобных объектов, а *экземпляр* – конкретный объект из этого множества. Например, тип объекта (тип сегмента) – стол, тогда экземпляр – тот конкретный стол, за которым вы работаете, описываемый его набором характеристик (цвет, размеры, материал и т.д.).

Родительской записи соответствует произвольное количество экземпляров подчиненных записей каждого типа. Для того чтобы можно было однозначно различать записи, каждый тип записи должен иметь *ключ* – набор полей, однозначно идентифицирующий экземпляр записи. Например, в записи, описывающей человека, таким ключом может быть номер паспорта. БД представляет собой набор таких деревьев.

*Связь* (англ. link) – иерархическое отношение между записями двух типов. Связи при графическом изображении обозначаются дугами ориентированного графа, типы записей – вершинами. Дуги графа, обозначающие логические связи, являются направленными. Запись, из которой выходит стрелка, является логически исходной; запись, в которую приходит стрелка, – логически подчиненной. Иногда исходные сегменты называют сегментами-предками, а подчиненные сегменты называют сегментами-потомками.

Тип связи определяется ее именем и задает свойства, общие для всех экземпляров связи данного типа. Экземпляр связи задается логически исходной записью («владельцем») и множеством (возможно пустым) подчиненных записей. Таким образом, каждой подчиненной записи в иерархической модели может соответствовать только одна исходная; одной исходной записи может соответствовать несколько подчиненных.

В *иерархической модели* сегменты и связи между ними создают древовидные структуры (деревья). В каждом дереве существует только одна запись, которая не связана ни с какой исходной записью, – она называется *корневой*. Таким образом, дерево – совокупность *корневой записи* и множества подчиненных записей.

*Схема иерархической БД* представляет собой совокупность отдельных деревьев, каждое дерево в рамках модели называется *физической базой данных*. Каждая физическая БД удовлетворяет следующим иерархическим ограничениям:

- в физической БД существует один *корневой сегмент*, т.е. сегмент, у которого нет логически исходного (родительского) типа сегмента;

- каждый логически исходный сегмент может быть связан с произвольным числом логически подчиненных сегментов;
- каждый логически подчиненный сегмент может быть связан только с одним логически исходным (родительским) сегментом.

Набор всех экземпляров записей, подчиненных одному экземпляру корневой записи, называется *физической записью*.

Экземпляры-потомки одного типа, связанные с одним экземпляром записи-предка, называются *близнецами*.

Количество экземпляров-потомков может быть разным для разных экземпляров родительских сегментов, поэтому в общем случае физические записи имеют разную длину. Физические записи в иерархической модели различаются по длине и структуре.

В примере на рис. 1.14, б записи-близнецы типа «Начальник отдела» – Петров и Сидорова, типа «Служащий» – Николаев и Васильев.



Рисунок 1.14. Иерархические базы данных:

а – пример структуры дерева; б – пример отдельной физической записи

Чтобы записи могли быть однозначно идентифицированы, применяется следующая схема. Корневая запись каждого дерева обязательно содержит следующую схему. Корневая запись каждого дерева обязательно содержит ключ с уникальными значениями. Ключи некорневых записей должны быть уникальны только среди «близнецов». Каждая запись идентифицируется полным составным ключом, под которым понимается совокупность ключей записей, начиная от корневой и далее вниз по иерархическому пути до искомой записи.

К каждой записи БД существует только один (иерархический) путь откорневой записи. Например, как видно из рисунка 1.15, для записи С4 путь проходит через записи А и В3.

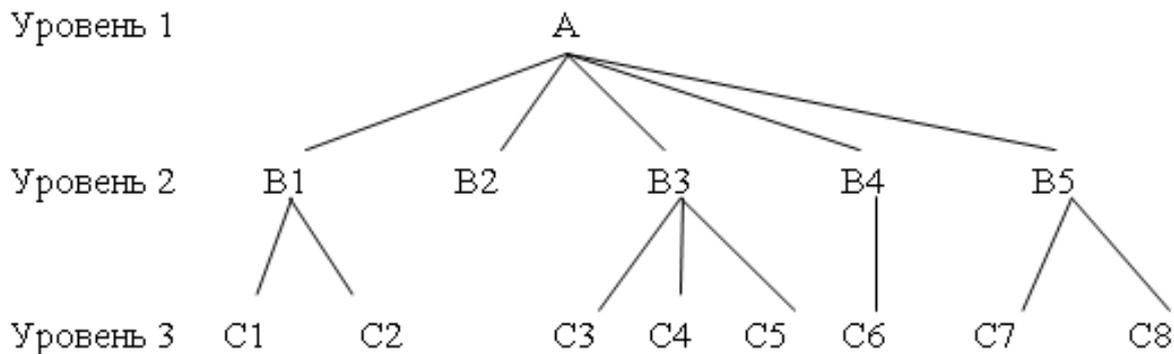


Рисунок 1.15. Графическое изображение иерархической структуры базы данных

На рисунке 1.16 представлен пример, который иллюстрирует использование иерархической модели БД. Для данного примера иерархическая структура правомерна, так как каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) институту.

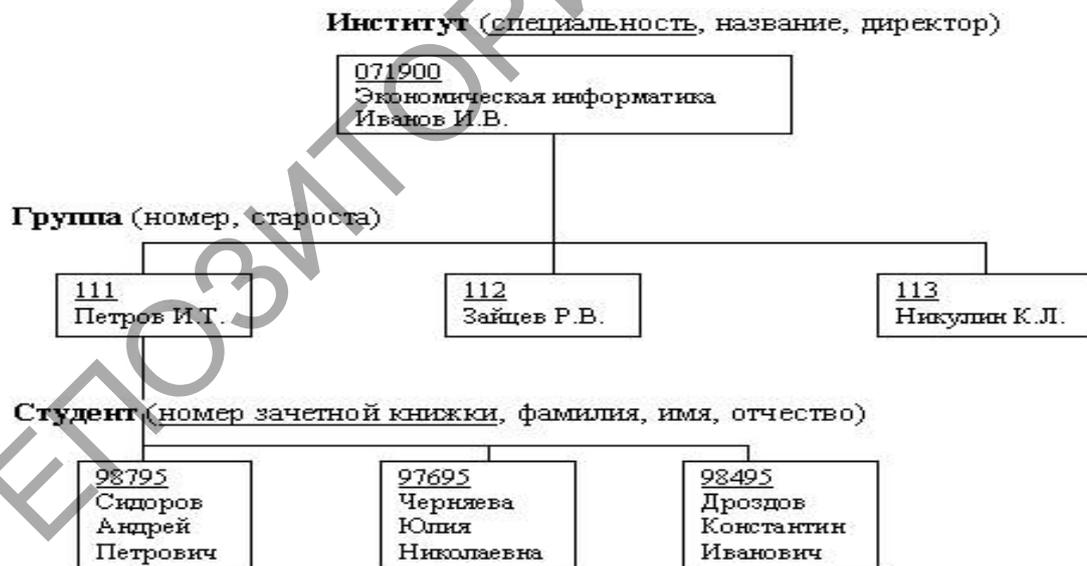


Рисунок 1.16. Пример иерархической структуры базы данных

Основной единицей работы в иерархической модели является запись, над ней могут производиться следующие операции:

- операция добавить (INSERT) позволяет занести в БД новую запись (с учетом ограничений, связанных с требованиями относительно уникальности ключа в рамках всей БД для корневой записи и среди «близнецов» – для

подчиненной);

- операция обновить (UPDATE) дает возможность изменить значения данных предварительно извлеченной записи. Ключевые данные записи не могут подвергаться обновлению, в подобных случаях старая запись должна быть удалена, а новая, содержащая измененные данные, – заполнена;

- операция удалить (DELETE) служит для исключения из БД некоторой записи и всех подчиненных ей;

- операция извлечь (GET) имеет несколько модификаций – извлечь по значению ключа, извлечь следующую запись (в порядке левостороннего обхода дерева), извлечь следующую, удовлетворяющую дополнительному условию и т.д.

Надо отметить, что в любом случае обработка начинается с корневой записи, и доступ к некорневым записям осуществляется по иерархическому пути.

Рассмотрим две модели (рис. 1.17), описывающие связь «преподаватель–студент». В первой (А) преподаватель выше по иерархии, во второй (Б) – студент. Первая полезна, когда нужно выяснить, кого учил преподаватель, вторая – когда студент хочет узнать, у кого он учился.

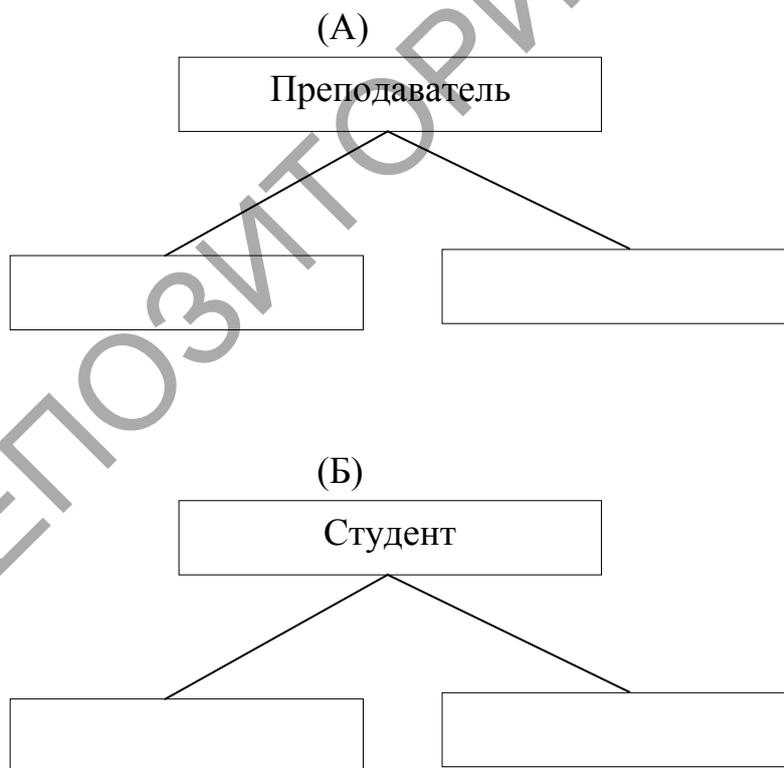


Рисунок 1.17. Иерархические модели связи «Преподаватель–Студент»

В этих моделях при работе с соответствующими БД возникают

нежелательные явления – *аномалии*.

*Аномалия включения:* В БД (А) невозможно включить студента, который пока ни у кого не обучался, в (Б) невозможно включить не обучающего преподавателя.

*Аномалия удаления:* так как при удалении сегмента исключаются и все его подчиненные, то при удалении преподавателя из (А) исчезают все его студенты, при удалении студента из (Б) – все его преподаватели.

На рисунке 1.18 демонстрируется своеобразный выход из положения: организуются две иерархические базы со ссылками из нижних уровней одной к верхнему уровню другой. Аномалий в этом случае нет, но работа с данными замедляется.

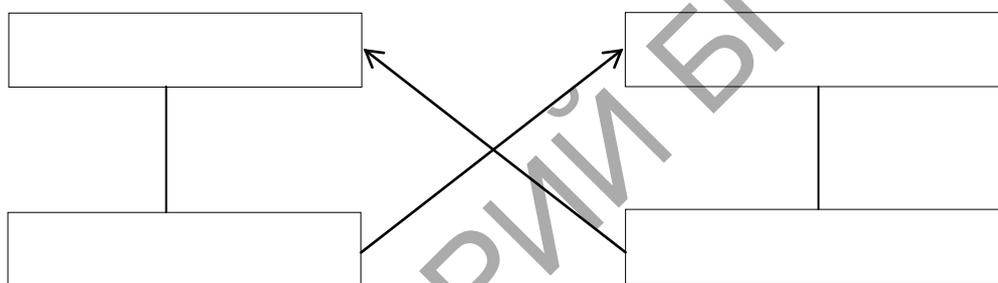


Рисунок 1.18. Взаимосвязанные иерархические модели связи «Преподаватель–Студент»

Достоинства модели:

- простота в понимании и использовании;
- обеспечение определенного уровня независимости по сравнению с файловыми системами;
- простота обслуживания и оценки характеристик, так как благодаря процедуре удаления нет нужды заботиться о висящих ссылках;
- высокая скорость выполнения основных операций над данными;
- удобство работы с иерархически упорядоченной информацией.

Недостатки модели:

- громоздкость такой модели для обработки информации с достаточно сложными логическими связями;
- трудность в понимании ее функционирования обычным пользователем;
- невозможность реализации отношения «многие ко многим» в рамках одной БД. Реализация такого отношения на основе двух БД затрудняет

управление

3. Сетевая модель данных: основные понятия, форма представления, достоинства и недостатки.

Сетевая модель может быть представлена как развитие и обобщение иерархической модели данных, позволяющее отображать разнообразные взаимосвязи данных в виде произвольного графа, обобщая тем самым иерархическую модель данных (рис. 1.19).

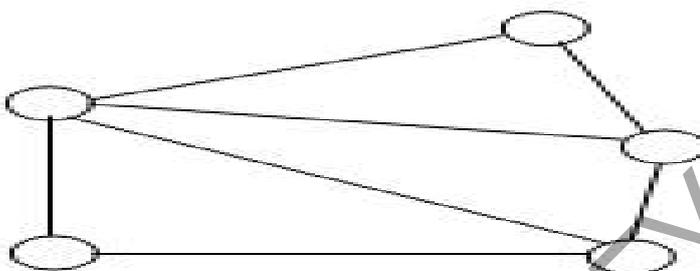


Рисунок 1.19. Представление связей в сетевой модели

Конференция по языкам систем данных (Conference on Data Systems Languages, CODASYL), поручила подгруппе, названной Database Task Group (DBTG), разработать стандарты систем управления базами данных.

Стандарт сетевой модели впервые был определен в 1975 году организацией CODASYL (Conference of Data System Languages). Так же как и иерархическая, сетевая модель относится к разряду теоретико-графовых, но она позволяет строить структуры данных, описываемые графом более общего вида, чем предполагает иерархическая модель.

Базовые структуры данных сетевой модели: элемент данных, агрегат данных, запись (или группа), набор (групповое отношение), БД (рис. 1.20).

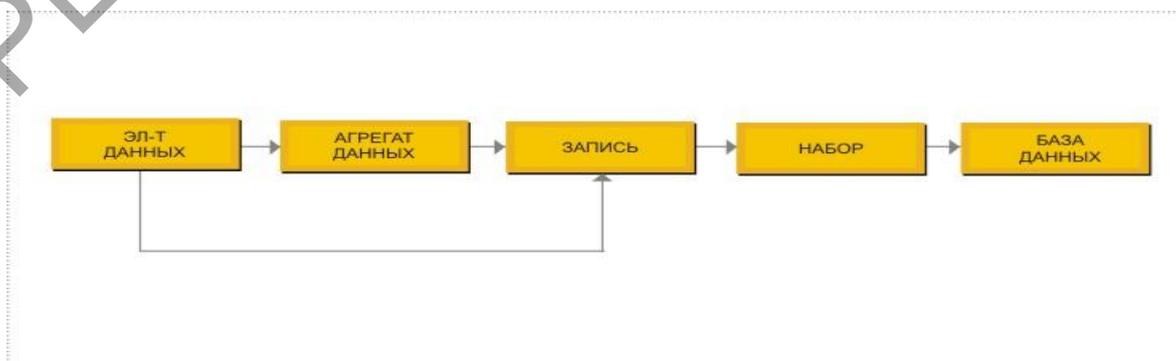


Рис. 1.20. Основные структуры сетевой модели данных

*Элемент данных*– это наименьшая поименованная информационная единица данных, доступная пользователю (аналог — поле в файловой системе). Элемент данных должен иметь свой тип (не структурный, простой).

*Агрегат данных* – именованная совокупность элементов или других агрегатов данных(рис. 1.21).Разница между элементом и агрегатом может быть проиллюстрирована следующим примером. Пусть в БД вносятся адреса. Если разработчик рассматривает адрес как единое целое (и соответствующим образом проектирует БД), то адрес – это элемент данных. Если же необходимо разделить адрес на части («страна» – «город» – «улица» – «номер дома» – «номер квартиры»), то адрес уже будет выступать как агрегат, состоящий из соответствующих элементов. При этом пользователь может запросить из БД как отдельно город или номер дома, так и адрес целиком, так как агрегат – это тоже именованный объект.

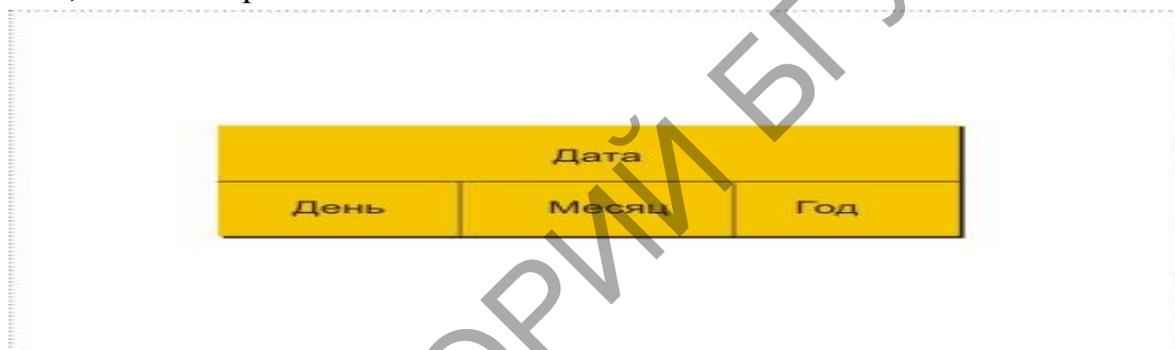


Рисунок 1.22. Агрегат Дата

Различают агрегаты типов «вектор» и «повторяющаяся группа». Агрегат, состоящий из простых элементов данных, называется вектором. Примером агрегата типа «вектор» может служить упомянутые выше адрес и дата, представленные как совокупности элементов. Агрегат, повторяющийся компонент которого представлен совокупностью данных, называется повторяющейся группой. В повторяющуюся группу могут входить элементы данных и другие агрегаты. В качестве примера здесь можно привести агрегат «Оценки студента», в котором будет сохраняться название предмета и оценка, причем столько раз, сколько экзаменов сдаст студент. Или в заказе может быть указано несколько видов товаров с числом повторений 10 (рис. 1.23).

Заказ						
Номер заказа	Дата заказа			Товар		
	День	Месяц	Год	Шифр товара	Кол-во товара	Наименование товара
Повторяющаяся группа						

Рис. 1.23. Запись Заказ

*Запись* – это агрегат, который не входит в состав никакого другого агрегата, и содержит один или более именованных элементов данных, каждый из которых обладает своим особым форматом. *Запись* обычно описывает некоторый объект реального мира и составляет основную единицу обработки БД (записи запоминаются, извлекаются, удаляются). Агрегат данных Дата входит в состав записи Сотрудник (рис. 1.24).

Тип записей – это совокупность логически связанных экземпляров записей. Тип записей моделирует некоторый класс объектов реального мира.

Сотрудник					
Табельный №	ФИО	Дата			Адрес
		День	Месяц	Год	

Рисунок 1.24. Запись Сотрудник

*Набор* – это поименованная двухуровневая иерархическая структура, которая содержит запись владельца и записи членов. Наборы выражают связи «один ко многим» или «один к одному» между двумя типами записей. Тип набора поддерживает работу с внутренними структурами типов записей. Набор, приведенный на рисунке 1.25, определяет тип записи-владельца Отдел и тип записи-члена набора Сотрудник, а также тип связи между ними «один ко многим» – с именем Работает. Имя набора – это метка, присвоенная

стрелке. Связь типа «один ко многим» допускает возможность того, что с данным экземпляром записи-владельца может быть связан ноль, один, или несколько экземпляров записи-члена.



Рисунок 1.25. Диаграмма типа набора Работает

Используя понятия сетевой модели данных, можно получить другое изображение такого набора (рис. 1.26), где представлены логические типы записей Отдел и Сотрудник, их структура и связь между типами записей Работает. *БД в сетевой модели данных – это поименованная совокупность экземпляров записей различного типа и экземпляров наборов, содержащих связи между ними.*



Рис. 1.26. Набор Работает между двумя типами записей Отдел и Сотрудник

В сетевой модели один и тот же тип записи может участвовать в

нескольких наборах. В частности, для любых двух типов записей может быть задано любое количество наборов, которые их связывают. Наличие подобных возможностей позволяет моделировать отношение объектов типа «многие-ко-многим», что выгодно отличает сетевую модель данных от иерархической.

В то же время, ни в иерархической, ни в сетевой модели один и тот же тип записи не может быть одновременно и владельцем, и членом группового отношения (набора).

На рисунке 1.27 с помощью диаграмм Бахмана проиллюстрировано различие между групповыми отношениями иерархической и сетевой моделей данных.



Рисунок 1.27. Особенности групповых отношений в сетевой и иерархической моделях данных

На диаграмме отображены отношения между типами записей «Человек», «Организация», «Поликлиника». В рамках сетевой модели можно описать, что между типами записей «Человек» и «Организация» имеется связь «многие-ко-многим»: один человек может работать в нескольких организациях, и в одной организации могут работать несколько человек. В иерархической модели в аналогичном случае придется ввести два типа записей «Организация» и «Организация\_1» и частично продублировать значения.

Для описания схемы сетевой БД используется две группы типов: «запись» и «связь». Тип «связь» определяется для двух типов «запись»: предка и потомка. Переменные типа «связь» являются экземплярами связей. Сетевая БД состоит из набора записей и набора соответствующих связей. На формирование связи особых ограничений не накладывается. Если в иерархических структурах запись-потомок могла иметь только одну запись-

предка, то в сетевой модели данных запись-потомок может иметь произвольное число записей-предков (сводных родителей).

Для манипулирования данными в сетевой модели данных определен ряд типичных операций, которые можно подразделить на две группы: *навигационные операции* и *операции модификации*. *Навигационные операции* осуществляют перемещение по БД путем прохождения по связям, определенным в схеме БД. В результате таких операций определяется запись, которая называется текущей. К подобным операциям относятся:

- найти конкретную запись в наборе однотипных записей и сделать ее текущей;
- перейти от записи-владельца к записи-члену в некотором наборе;
- перейти к следующей записи в некоторой связи;
- перейти от записи-члена к владельцу по некоторой связи.

*Операции модификации* осуществляют как добавление новых экземпляров отдельных типов записей, так и экземпляров новых наборов, удаление экземпляров записей и наборов, модификацию отдельных компонентов самой записи. Для реализации этих операций в системе текущее состояние детализируется путем запоминания трех его составляющих: текущего набора, текущего типа записи, текущего экземпляра типа записи. В такой ситуации возможны следующие операции:

- извлечь текущую запись в буфер прикладной программы для обработки;
- заменить в извлеченной записи значения указанных элементов данных на заданные новые их значения;
- запомнить запись из буфера в БД;
- создать новую запись;
- уничтожить запись;
- включить текущую запись в текущий экземпляр набора;
- исключить текущую запись из текущего экземпляра набора.

Сетевая модель универсальна и по сравнению с иерархической имеет гораздо большие возможности по моделированию связей между объектами.

Основные достоинства сетевой модели состоят в том, что:

- отсутствует дублирование данных в различных объектах модели;
- доступ к данным практически не имеет ограничений и возможен непосредственно к объекту любого уровня;
- допустимы всевозможные запросы;
- допустимо образование произвольных связей.

Основными недостатками сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также сложность для понимания и выполнения обработки информации в БД обычным пользователем. Кроме того, в сетевой модели данных ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.

В сетевых БД сложно проводить изменение структуры, так как приходится перестраивать множество различных связей. Сетевую модель трудно реализовывать, так как программист должен знать все связи между разнородными объектами. Построение приложений является трудоемким процессом.

Первая СУБД, построенная по сетевой модели – IDMS (1971 г.). Правами на нее обладает компания Computer Associates, она до сих пор поставляет и развивает эту СУБД. Примером может служить и СУБД IMAGE/1000 фирмы Hewlett-Packard.

4. Реляционная модель данных: основные понятия, форма представления, достоинства и недостатки.

Реляционная модель данных в настоящее время используется в большинстве коммерческих СУБД.

Сотрудник фирмы IBM доктор Э.Ф. Кодд, математик по образованию, в 1970 году предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение). Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как *отношение* (relation).

Положив *теорию отношений* в основу реляционной модели, Э.Ф. Кодд обосновал реляционную замкнутость отношений и ряда некоторых специальных операций, которые применяются сразу ко всему множеству строк отношения, а не к отдельной строке. Указанная реляционная замкнутость означает, что результатом выполнения операций над отношениями является также отношение, над которым в свою очередь можно осуществить некоторую операцию. Из этого следует, что в данной модели можно оперировать реляционными выражениями, а не только отдельными операндами в виде простых имен таблиц.

Предложенная Э.Ф.Коддом реляционная модель позволила пользователю не заботиться о физической структуре данных и не

интересоваться ею. Кодд ввел два языка манипулирования данными – *реляционная алгебра* и *реляционное исчисление*, которые предлагали более эффективные средства доступа к данным и их обработки и в настоящее время являются основой коммерческих СУБД.

Представление данных в виде множественной совокупности таблиц позволило избежать многих недостатков ранних СУБД и создать системы с упрощенным интерфейсным управлением. Многие серверы баз данных основаны на принципах работы, связанных с реляционной моделью.

К числу достоинств *реляционного подхода* можно отнести: □

– наличие небольшого набора абстракций, которые позволяют сравнительно просто моделировать большую часть распространенных предметных областей и допускают точные формальные определения, оставаясь интуитивно понятными; □

– наличие простого и в то же время мощного математического аппарата, опирающегося главным образом на теорию множеств и математическую логику и обеспечивающего теоретический базис реляционного подхода к организации БД;

– возможность ненавигационного манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти.

Популяризатор идей Э.Ф. Кодда Кристофер Дейт, который воспроизводит ее (с различными уточнениями) практически во всех своих книгах (например, К. Дейт. Введение в системы баз данных. – 6-е изд. – М.; СПб.: Вильямс, 2000), в реляционной модели выделяет три части, описывающие разные аспекты реляционного подхода:

- 1) структурная часть;
- 2) манипуляционная часть;
- 3) целостная часть.

*Структурную часть* реляционной модели составляют следующие компоненты:

– *отношения неопределенного порядка*, концептуально представлены таблицами;

– *атрибуты* – атомарные данные, характеризующие отношения и представлены столбцами таблицы;

– *домены* – множества допустимых значений атрибутов;

– *кортежи* – совокупности значений всех атрибутов отношения, взятых по одному для каждого атрибута, представлены строками таблицы;

– *возможные ключи* – множество атрибутов, однозначно

определяющее кортеж в отношении;

– *первичные ключи* – для каждого отношения это один из возможных ключей.

*Манипуляционная часть* состоит из операторов выбора, проекции, соединения и т.п., которые преобразуют отношения в отношения.

Целостная часть состоит из правил: правил целостности на уровне *объектов* и правил целостности на уровне *ссылок*. В любой реализации можно определить ограничения, которые определяют меньшее множество возможных непротиворечивых значений.

Логическая структура данных представляется набором связанных таблиц. Модель поддерживает связи «один к одному» и «один ко многим». Связь «многие ко многим» реализуется с помощью декомпозиции, то есть легко разрешается введением дополнительного (производного) отношения, которое является промежуточным между исходными отношениями и соединено с ними двумя связями «один ко многим».

Основные элементы реляционной модели данных представлены в рисунке 1.28.

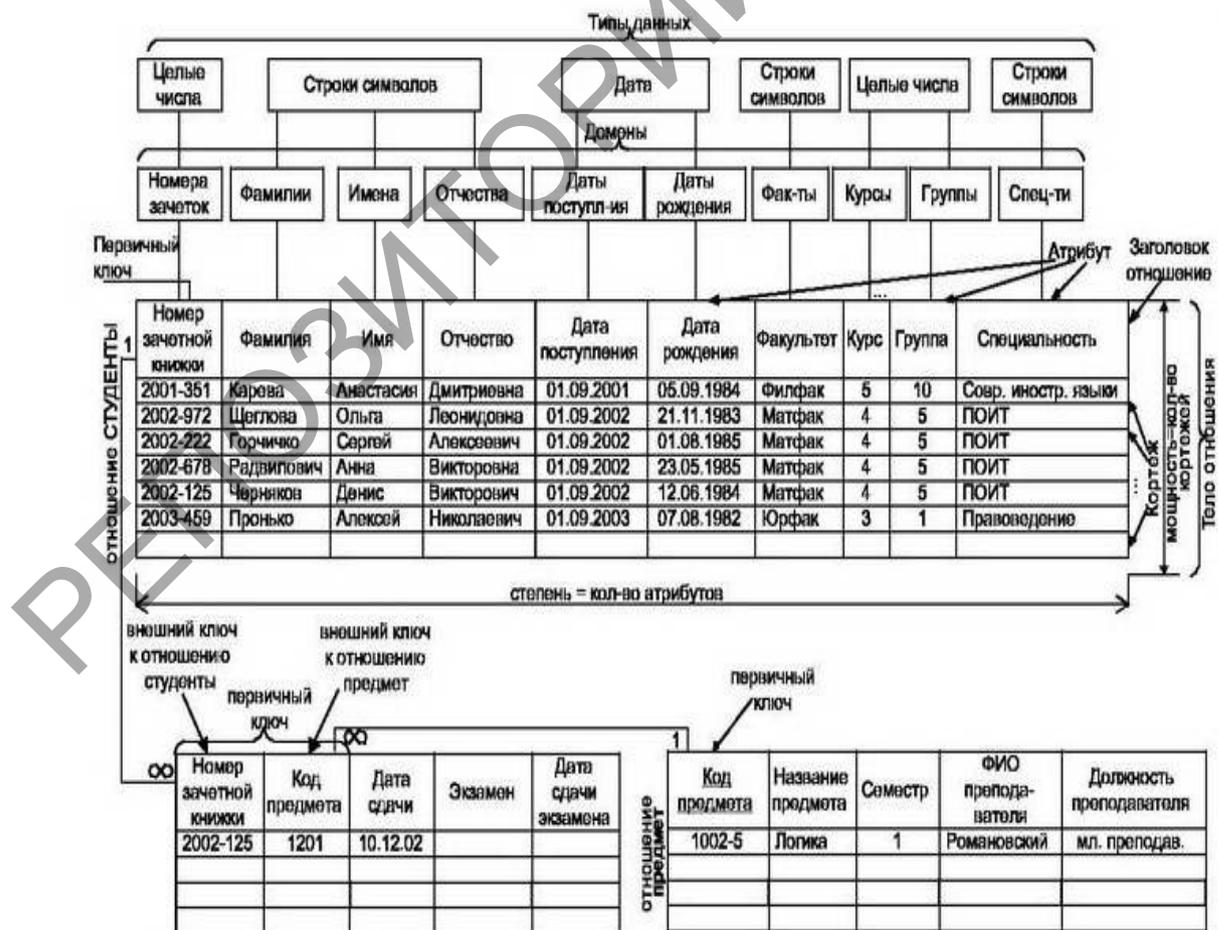


Рисунок 1.28. Элементы реляционной модели данных

Краткие описания основных понятий реляционной модели приведены в

таблице 1.

Таблица 5.1. Основные понятия реляционной модели

Термин реляционной модели	Описание
База данных (БД)	Набор таблиц и других объектов, необходимых для абстрактного представления выбранной предметной области
Схема БД	Набор заголовков таблиц, взаимосвязанных друг с другом
Отношение	Таблица – совокупность объектов реального мира, которые характеризуются общими свойствами и характеристиками (поля таблицы)
Заголовок отношения	Заголовок таблицы – названия полей (столбцов) таблицы
Тело отношения	Тело таблицы – совокупность значений для всех объектов реального мира, которая представима в виде записей таблицы (строки таблицы)
Схема отношения	Строка заголовков столбцов таблицы («шапка» таблицы)
Атрибут отношения	Наименование столбца таблицы (поле таблицы)
Кортеж отношения	Строка таблицы (запись) – однозначное представление объекта реального мира, созданное с использованием значений полей таблицы
Домен	Множество допустимых значений атрибута
Значение атрибута	Значение поля в записи (кортеже)
Первичный ключ	Один или несколько (в случае составного ключа) атрибутов, которые единственным образом определяют значение кортежа (значение строки таблицы)
Внешний ключ	Атрибут таблицы, значения которого соответствуют значениям первичного ключа в другой связанной (родительской, первичной) таблице. Внешний ключ может состоять как из одного, так и из нескольких атрибутов (составной внешний ключ). В случае если число атрибутов внешнего ключа меньше, чем количество атрибутов соответствующего первичного ключа, то он принято называть усеченным (частичным) внешним ключом
Степень (арность) отношения	Количество столбцов таблицы
Мощность отношения	Количество строк (кортежей) таблицы

Экземпляр отношения	Множество записей (кортежей) для данной таблицы (отношения). С течением времени экземпляр может изменяться. Поскольку обычная БД в текущий момент времени работает только с одной версией отношения, то такой экземпляр отношения принято называть текущим
Тип данных	Тип значений элементов таблицы
Базовое отношение	Отношение, содержащие один или несколько столбцов, характеризующих свойства объекта, а также
Производное отношение	Не является базовым отношением, т.е. не характеризует свойства объекта и используется для обеспечения связей между другими таблицами, может не содержать первичного ключа. В случае если первичный ключ задан, то он состоит из внешних ключей, связанных с первичными ключами базового отношения
Связь	Устанавливает взаимосвязь между совпадающими значениями в ключевых полях – первичным ключом одной таблицы и внешним ключом другой
Связь «один-к-одному» (1:1)	При использовании этого вида связи запись в одной таблице может иметь не более одной связанной с ней записи в другой таблице. В обеих таблицах ключевые поля должны быть первичными. Используется для разделения таблиц с многочисленными полями или по требованию защиты данных
Связь «один-ко-многим» (1:M)	При использовании этого вида связи каждой записи одной таблицы может соответствовать несколько записей второй, но каждой записи второй таблицы соответствует лишь одна запись первой таблицы. В первой таблице обязательно должен быть задан первичный ключ, во второй – внешний
Связь «многие-ко-многим» (N:M)	При данном типе связи одной записи в первой таблице может соответствовать несколько записей второй таблицы, но и одной записи второй таблицы может соответствовать несколько записей первой. Уникальность ключей для таких таблиц не требуется. В процессе проектирования схемы БД такие связи преобразуют. Для этого крайне важно ввести вспомогательное отношение, позволяющее заменить связь «многие-ко-многим» на две связи типа «один-ко-многим»

Э.Ф. Кодд приводит критерии, по которым СУБД можно отнести к реляционным. Эти критерии следующие:

– СУБД должна поддерживать таблицы без видимых пользователю навигационных связей;

– язык манипулирования данными должен обеспечивать минимальную возможность реляционной обработки, то есть включать операторы выбора, проекции и соединения.

Если СУБД не удовлетворяет второму критерию, ее называют *табличной (полуреляционной)*. Реляционные СУБД с минимальной возможностью реляционной обработки называются *минимальнореляционными*, если же в

СУБД в полной мере реализованы две последние составляющие реляционной модели, – это *полностью реляционные СУБД*. СУБД, реализующие полный набор реляционных операторов, называются *реляционно полными*.

Пусть имеется отношение  $r$ . Схемой отношения  $r$  называется конечное множество имен атрибутов  $R = \{A_1, A_2, \dots, A_n\}$ . Заголовки столбцов отношения содержат имена его атрибутов и, следовательно, все вместе отражают его схему. Схема отношения ПРЕПОДАВАТЕЛЬ может быть представлена следующим образом:

{Таб\_ном\_преп, Фамилия, Должность}

или

ПРЕПОДАВАТЕЛЬ
<u>Таб_ном_преп</u>
Фамилия
Должность

Тогда заголовок отношения ПРЕПОДАВАТЕЛЬ примет вид

Таб_ном_преп	Фамилия	Должность
--------------	---------	-----------

Отношение строится с учетом ряда факторов. Каждому имени атрибута  $A_i$ ,  $1 \leq i \leq n$  ставится в соответствие множество допустимых для соответствующего столбца значений. Это множество  $D_i$ , называется доменом данного имени атрибута. Каждая строка отношения является множеством значений, взятых по одному из домена каждого имени атрибута. Домены являются произвольными непустыми конечными или счетными множествами и образуют множество:  $D = D_1 \cup D_2 \cup \dots \cup D_n$ . Отношение  $r$  со схемой  $R$  – это конечное множество отображений  $\{t_1, t_2, \dots, t_p\}$  из  $R$  в  $D$ . Причем каждое отображение  $t \in r$  должно удовлетворять следующему ограничению:  $t(A_i)$  принадлежит  $D_i$  где  $1 \leq i \leq n$ . Эти отображения называются кортежами. Каждый кортеж отношения отображает экземпляр сущности, а атрибут отношения отображает атрибут сущности. Множество кортежей называется телом отношения. Тело отношения отражает состояние сущности, поэтому во времени оно постоянно меняется. Тело отношения характеризуется кардинальным числом, которое равно количеству содержащихся в нем кортежей. Одной из главных характеристик отношения является его степень. Степень отношения определяется количеством атрибутов, которое в нем присутствует. Эта характеристика отношения имеет еще названия: ранг и

арность. Отношение с одним атрибутом называется унарным, с двумя атрибутами – бинарным, с тремя – тернарным, с n атрибутами n-арным. Определение степени отношения осуществляется по заголовку отношения.

Все названные характеристики отношения обозначены на рисунке 1.29.



Рисунок 1.29. Отношение СОТРУДНИК

Отношение по структуре подобно таблице, но таблице, обладающей определенными свойствами. Свойства отношения следующие:

1. Отношение имеет имя, которое отличается от имен всех других отношений. □
2. Отношение представляется в виде табличной структуры.
3. Каждый атрибут имеет уникальное имя, его значения берутся из одного и того же домена.
4. Каждый компонент кортежа является простым, атомарным значением, не состоящим из группы значений.
5. Упорядочение атрибутов теоретически несущественно, однако оно может влиять на эффективность доступа к кортежам. □
6. Все строки (кортежи) должны быть различны.
7. □ Теоретически порядок следования кортежей не имеет значения.

В отношении могут существовать несколько одиночных или составных атрибутов, которые однозначно идентифицируют кортеж отношения. Это – потенциальные ключи.

Говорят, что множество атрибутов  $K = \{A_i, A_j, \dots, A_k\}$  отношения  $r$  является потенциальным ключом  $r$  тогда и только тогда, когда

удовлетворяются два независимых от времени условия:

- уникальность: в произвольный заданный момент времени никакие два различных кортежа  $r$  не имеют одного и того же значения для  $A_i, A_j, \dots, A_k$ ;
- минимальность: ни один из атрибутов  $A_i, A_j, \dots, A_k$  не может быть исключен из  $K$  без нарушения уникальности.

Отношение может иметь несколько потенциальных ключей. Ключ, содержащий два и более атрибута, называется *составным ключом*. Каждое отношение обладает хотя бы одним возможным ключом, поскольку в отношении не может быть одинаковых кортежей, а это значит, что, по меньшей мере, комбинация всех его атрибутов удовлетворяет условию уникальности. Потенциальные ключи, позволяя гарантированно выделить точно один кортеж, обеспечивают основной механизм адресации на уровне кортежей реляционной модели.

Один из возможных ключей (выбранный произвольным образом) принимается за его *первичный ключ*. Обычно первичным ключом назначается тот возможный ключ, которым проще всего пользоваться при повседневной работе. Остальные возможные ключи, если они есть, называются *альтернативными ключами*. Для индикации связи между отношениями используются *внешние ключи*.

*Внешний ключ* – это набор атрибутов одного отношения, являющийся потенциальным ключом другого отношения.

Благодаря наличию связей между потенциальными и внешними ключами обеспечивается взаимосвязь кортежей определенных отношений.

Отношение, содержащее внешний ключ, называется *дочерним или ссылающимся отношением*. А отношение, содержащее связанный с внешним ключом потенциальный ключ, — *родительским или целевым отношением*.

Отношения не могут рассматриваться как статические объекты, так как они предназначены для отражения некоторой части реального мира, а эта часть реального мира может изменяться во времени. Поэтому и отношения изменяются во времени: кортежи могут добавляться, удаляться или модифицироваться. Тем не менее, предполагается, что сама схема отношения инвариантна во времени. Отношение должно восприниматься как множество возможных состояний, которые может принимать отношение.

#### Пример

Пусть рассматривается концептуальная модель, приведенная на рисунке 1.30. Она относится к предметной области, которую можно назвать «Преподавательская деятельность». Данная модель содержит две сущности: ЛЕКТОР и ПРЕДМЕТ, между которыми установлена связь ЧИТАЕТ

типа «многие ко многим». Характеристики сущностей представлены изображенными на рисунке атрибутами. Связь ЧИТАЕТ не имеет собственных атрибутов.

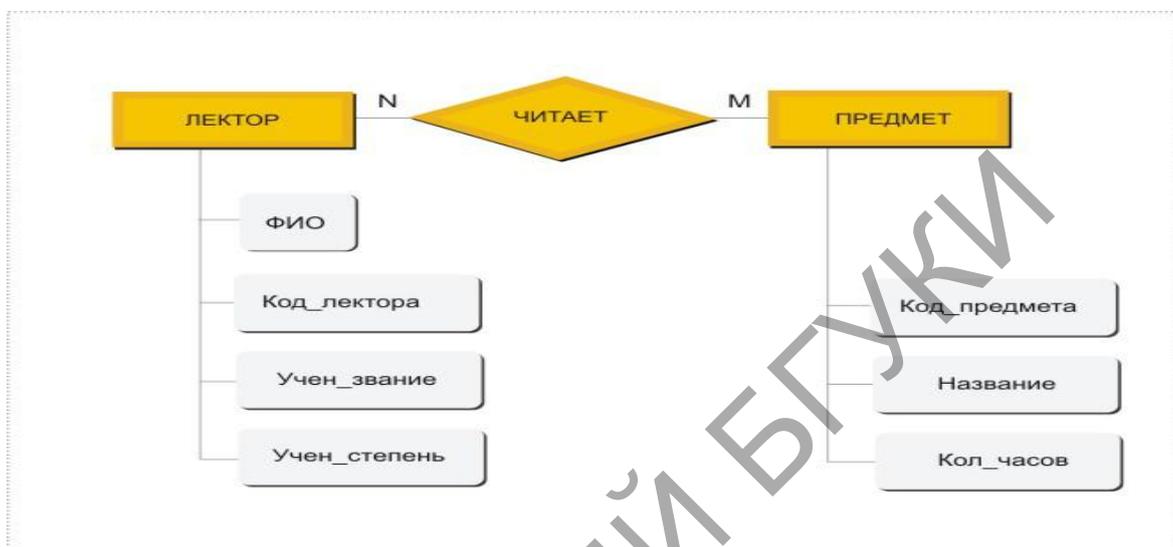


Рисунок 1.30. Концептуальная модель

Реляционная схема, соответствующая указанной концептуальной модели включает в себя три отношения: ЛЕКТОР, ПРЕДМЕТ, ЧИТАЕТ. Схемы отношений и связи между ними изображены на рисунке 1.31.

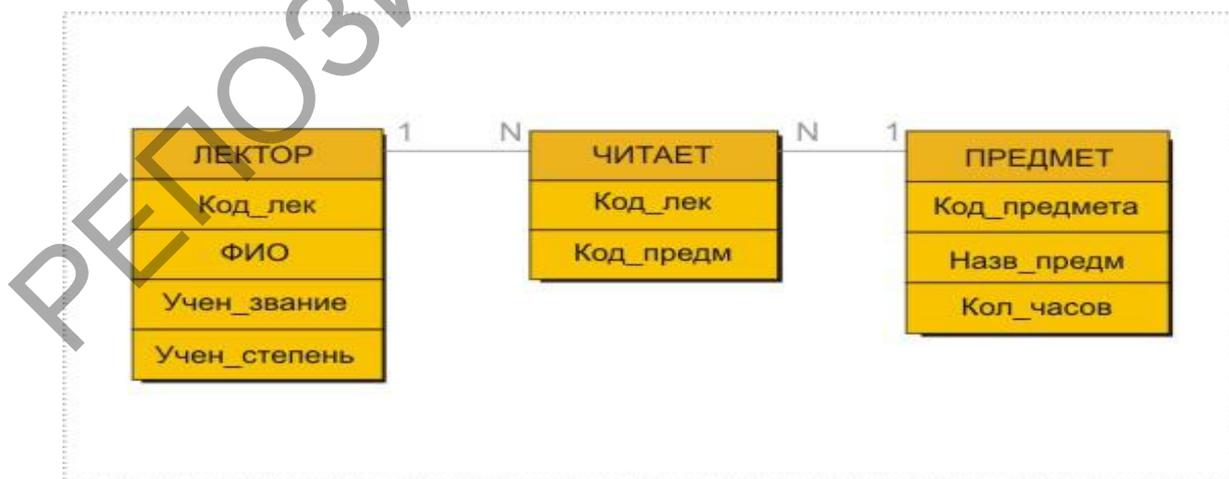


Рисунок 1.31. Реляционная схема базы данных

На рисунке 1.32 даны соответствующие отношения, заполненные кортежами. Эти отношения имеют следующие характеристики:

ЛЕКТОР– 4-арное отношение с первичным ключом Код\_лек кардинальным числом, равным четырем; атрибуты определены на следующих доменах: Код\_лек– {целые: 1...4}, ФИО – {возможные фамилии и инициалы}, Уч\_степень– {к.т.н., д.т.н., нет степени}, Уч\_звание– {Доцент, Профессор, Нет\_звания};

ПРЕДМЕТ–тернарное отношение с первичным ключом Код\_предм с кардинальным числом, равным шести; атрибуты определены на следующих доменах: Код\_предм– {символьный}. Назв\_предм– {Информатика, Программирование, Физика, ООП, Базы данных, Базы данных}, Кол\_во\_час — {целые: 54, 102, 36};



Рис. 1.32. Реляционные отношения модели

Все приведенные отношения являются нормализованными, поскольку атрибуты всех отношений имеют атомарные значения. Во всех трех отношениях отсутствуют дублирующие кортежи.

Для обновления отношений необходимо иметь возможность выполнять следующие операции:

- добавление кортежа;
- удаление кортежа;
- изменение кортежа.

Операция добавления для отношения  $r$  со схемой  $(A_1, A_2, \dots, A_n)$  имеет вид:

$ADD (r: A_1=d_1, A_2=d_2, \dots, A_n=d_n).$

Если порядок атрибутов фиксирован, возможна более короткая запись:

$ADD (r: d_1, d_2, \dots, d_n).$

Выполнение этой операции может стать невозможным в ряде случаев:

- добавляемый кортеж не соответствует схеме определенного отношения;
- некоторые значения кортежей не принадлежат соответствующим доменам;
- описанный кортеж совпадает по ключу с кортежем, уже находящимся в отношении.

Операция удаления предназначена для удаления кортежей. Она может быть записана следующим образом:

$DEL (r: A_1=d_1, A_2=d_2, \dots, A_n=d_n),$

или для упорядоченных атрибутов:

$DEL (r: d_1, d_2, \dots, d_n) .$

Для удаления некоторого кортежа часто достаточно указать значение некоторого ключа:

$DEL (r; \text{ключ}).$

Если указанный кортеж в отношении отсутствует, то отношение остается неизменным.

Операция изменения предназначена для модификации части кортежа. Для отношения  $r$  ее можно при  $\{C_1, C_2, \dots, C_p\} \in \{A_1, A_2, \dots, A_n\}$  определить так:

$CH (r: A_1=d_1, A_2=d_2, \dots, A_n=d_n; C_1=e_1, C_2=e_2, \dots, C_p=e_p).$

Если  $K = \{B_1, B_2, \dots, B_k\}$  является ключом, то запись данной операции может быть сокращена:

$CH (r: B_1=d_1, B_2=d_2, \dots, B_k=d_k; C_1=e_1, C_2=e_2, \dots, C_p=e_p).$

Возможные ошибки в данном случае те же, что и у предыдущих операций:

- указанный кортеж не существует;
- изменения имеют неправильный формат;
- используемые значения не принадлежат соответствующим доменам

Достоинствами реляционной модели являются простота, наглядность, независимость от данных. К тому же, в отличие от сетевых и иерархических моделей, реляционные модели для организации связей между записями применяют не внутренние указатели, а фактические значения атрибута,

используя общий атрибут в каждой из записей. Недостатки реляционной модели связаны с однородностью структуры данных, семантической перегруженностью модели, ограниченным набором операций.

Примерами реляционных СУБД являются следующие:

- dBase III Plus и dBase IV (фирма Ashton-Tate);
- DB2 (IBM);
- FoxPro ранних версий и FoxBase (Fox Software);
- Paradox и dBASE for Windows (Borland);
- Visual FoxPro и Access (Microsoft);
- Oracle (Oracle).

РЕПОЗИТОРИЙ БГУКИ

### **3. ПРАКТИЧЕСКИЙ РАЗДЕЛ**

#### **3.1 Методические указания к практическим и лабораторным занятиям**

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала предмета путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу.

Лабораторные работы представляют одну из форм освоения теоретического материала с одновременным формированием практических навыков в Их назначение – углубление проработки теоретического материала, формирование практических навыков путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к лабораторным работам включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу.

### 3.2 Тематика практических и лабораторных занятий

#### Практическая работа № 1. Классификация баз данных библиотек Республики Беларусь

##### Задание:

1. Провести классификацию баз данных следующих библиотек Республики Беларусь:

- Президентская библиотека Республики Беларусь;
- Центральная научная библиотека имени Якуба Коласа Национальной академии наук Беларуси;
- Белорусская сельскохозяйственная библиотека имени И.С. Лупиновича Национальной академии наук Беларуси;
- Республиканская научная медицинская библиотека;
- Республиканская научно-техническая библиотека.

##### Методика выполнения:

1. Запустить любой браузер сети Интернет.
2. Зайти на страницы библиотек, указанных в задании, в сети Интернет.
3. Провести анализ информации о базах данных, размещенной на страницах библиотек. Отобрать 10 баз данных, которые генерируются или приобретаются библиотеками, и осуществить их классификацию по ряду признаков: а) форма представления данных; б) вид данных; в) структуризация данных; г) характер данных; д) организация хранения; д) тип хранимой информации; е) функциональное назначение; ж) сфера возможного применения; з) степень доступности; и) количество пользователей.
4. Результаты анализа баз данных отразить в таблице 1 «Классификация баз данных библиотек Республики Беларусь».

Таблица 1. Классификация баз данных библиотек Республики Беларусь.

Название библиотеки и ее URL в сети Интернет										
Название базы данных	Вид базы данных по признаку									
	Формы представления данных	Вида данных	Структуриза ции данных	Характер а данных	Организации хранения	Типа хранимой информации	Функциональ ного назначения	Сферы возможного применения	Степени доступности	Количества пользователей

РЕПОЗИТОРИЙ БГУКИ

## Практическая работа № 2. Анализ систем управления базами данных.

### Задание:

1. Провести анализ технических и функциональных возможностей предложенных систем управления базами данных:

– Microsoft SQL Server –

[https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server);

– PostgreSQL – <https://www.postgresql.org/>;

– dBase – <https://ru.wikipedia.org/wiki/DBase>;

– Visual FoxPro – [https://ru.wikipedia.org/wiki/Visual\\_FoxPro](https://ru.wikipedia.org/wiki/Visual_FoxPro);

– MS Access – [https://ru.wikipedia.org/wiki/Microsoft\\_Access](https://ru.wikipedia.org/wiki/Microsoft_Access);

– MariaDB – <https://ru.wikipedia.org/wiki/MariaDB>;

– DB2 – <https://ru.wikipedia.org/wiki/DB2>;

– HANA – <https://ru.wikipedia.org/wiki/Hana>;

– Informix – <https://ru.wikipedia.org/wiki/Informix>;

– Oracle Database – <https://www.oracle.com/database/>;

– Firebird – <https://ru.wikipedia.org/wiki/Firebird>;

– MySQL – <https://www.mysql.com/>; <https://ru.wikipedia.org/wiki/MySQL>;

– Линтер –

<https://www.linter.ru/ru/https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D1%82%D0%B5%D1%80>;

– РЕД База Данных –

[https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94\\_%D0%A0%D0%B5%D0%B4\\_%D0%91%D0%B0%D0%B7%D0%B0\\_%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D1%85](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94_%D0%A0%D0%B5%D0%B4_%D0%91%D0%B0%D0%B7%D0%B0_%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

### Методика выполнения:

1. Запустить любой браузер сети Интернет.  
2. Зайти на страницы систем управления базами данных, предложенные в задании.

3. Провести анализ каждой системы управления базами данных по ряду параметров: а) название СУБД; б) компании-разработчики СУБД; в) операционная система СУБД; г) лицензия СУБД; д) исходный код СУБД (открытый/закрытый); е) тип СУБД по модели данных; ж) поддержка СУБД (платная/бесплатная); з) функциональные возможности СУБД.

4. Результаты анализа систем управления базами данных отразить в таблице 2 «Параметры систем управления базами данных».

## Литература

1. Панченко, И. PostgreSQL: вчера, сегодня, завтра / И. Панченко // Открытые системы. СУБД. – 2015. – № 3. – С. 34–37.
2. ЛИНТЕР как СУБД для интеллектуальных информационных систем. Часть 1 // Нейрокомпьютеры: разработка, применение. – 2009. – № 2. – С. 4–11.
3. СУБД ЛИНТЕР для интеллектуальных информационных систем. Часть 2, 3 / /С.П. Маркин, В.Л. Борисов, А.В. Распопов, В.Е. Максимов / Нейрокомпьютеры: разработка, применение. – 2009. – № 2. – С. 12–29.
4. Симаков, Р. СУБД «Ред База Данных» для серверных ОС / Р. Симаков // Системный администратор. – 2015. – № 7–8 (152–153). – С. 108–109.

Таблица 2. Параметры систем управления базами данных.

Название СУБД						
Компании-разработчики СУБД	Операционная система СУБД	Лицензия СУБД	Исходный код СУБД (открытый/закрытый)	Тип СУБД по модели данных	Поддержка СУБД (платная/бесплатная)	Функциональные возможности СУБД

РЕПОЗИТОРИЙ БГУКИ

**Практическая работа № 3. Выявление основных понятий реляционной модели данных.**

**Задание:**

1. Выявить основные понятия реляционной модели данных в предложенных таблицах (отношениях):

а)

Студенты	инструменты				факультеты			
	сакса-фон	пиано	ударные	контрабас	история	география	биология	физика
Андрей	0	1	0	1	0	0	0	1
Валерий	0	1	0	0	0	1	0	0
Михаил	1	0	0	0	1	0	0	0
Леонид	0	0	1	0	0	0	1	0

б)

Поставщики	Потребители				Запасы продукции	Индексы, $U$
	1	2	...	$n$		
1	$c_{11}$	$c_{12}$	...	$c_{1n}$	$a_1$	
2	$c_{21}$	$c_{22}$	...	$c_{2n}$	$a_2$	
...	...	...	...	...	...	
$m$	$c_{m1}$	$c_{m2}$	...	$c_{mn}$	$a_m$	
Потребности в продукции	$b_1$	$b_2$	...	$b_n$		
Индексы $V$						

в)

Подъем	Ряд					
	Передний		Средний		Задний	
	Нелабиализованный	Лабиализованный	Нелабиализованный	Лабиализованный	Нелабиализованный	Лабиализованный
Верхний	<b>i</b>	<b>y</b>	<b>ɨ</b>	<b>ɥ</b>	<b>ɯ</b>	<b>ɰ</b>
Средний	<b>e</b> <b>ɛ</b>		<b>ɛ</b> <b>e</b>	<b>ɵ</b>	<b>ɣ</b> <b>ʌ</b>	<b>o</b> <b>ɔ</b>
Нижний			<b>a</b>	<b>œ</b>	<b>a</b>	<b>ɒ</b>

г)

Ф.И.О.	Оценки по предметам			Суммарный балл успеваемости
	философия	психология	статистика	
Иванов П.А.	4	5	3	12
Петров И. И.	4	5	5	14
Сидоров П.П.	5	5	5	15
Николаев Д.А.	3	5	3	11
Федорова И.А.	3	4	5	12
Лазарева Л. В.	4	4	3	11
Средний балл успеваемости	3,33	4,67	4	xxxxxx

д)

№ п/п	Наименование Дисциплины и видов учебной работы студента	Распределение по семестрам		Объем работы (количество часов) С преподавателем						
		экзамен	зачет	Всего (часов)	Аудиторные занятия					Самостоятельная работа
					Всего	Занятия - лекции	Лабораторные занятия	Практические занятия	Индивидуальные занятия	
1	Информатика	2	1	232	136	34	102	-	30	66
2	Теория управления	6	5	300	200	50	50	50	50	100
Всего (ч):				532	336	84	152	50	80	166

е)

Название страны	Население млн. чел.		Плотность чел. на кв. км.		Площадь, млн. кв. км.
	1970 г	1989 г	1970 г	1989 г	
Австралия и Океания	19	26	2	3	8,5
Африка	361	628	12	21	30,3
Европа	642	701	61	67	10,5
Южная Америка	190	291	11	16	17,8
Северная и Центральная Америка	320	422	13	17	24,3
Азия	2161	3133	49	71	44,4
Весь мир	3693	5201	27	38	135,8

ж)

Буквенное обозначение	Нормированное произношение	Искаженное произношение	Замены
А	[a]	[ɐ], [œ], [ɑ], [ɒ], [ã]	—
О	[o]	[ɤ], [θ], [ə], [õ]	[u], [e]
И	[i]	[y], [ɪ], [ĩ]	[i]
У	[u]	[ɯ], [ũ]	[o]
Э	[e]	[ə], [ɛ], [ɔ], [ʌ], [ē]	—
Ы	[ɨ]	[ɯ], [ɪ], [ə], [ĩ]	[e], [i]

**Методика выполнения:**

1. Проанализировать структуру и состав таблиц.
2. На примере каждой таблицы выявить следующие основные понятия реляционной модели:
  - Тип данных (числовой, вещественный; строковый; денежный; дата; дата и время и др.);
  - Сущность;
  - Атрибут;
  - Отношение;
  - Домен (поле);
  - Кортеж (запись);
  - Схема отношения (строка заголовков);
  - Первичный ключ отношения.
3. Результаты работы отразить в виде по примеру, представленном на рисунке 1.

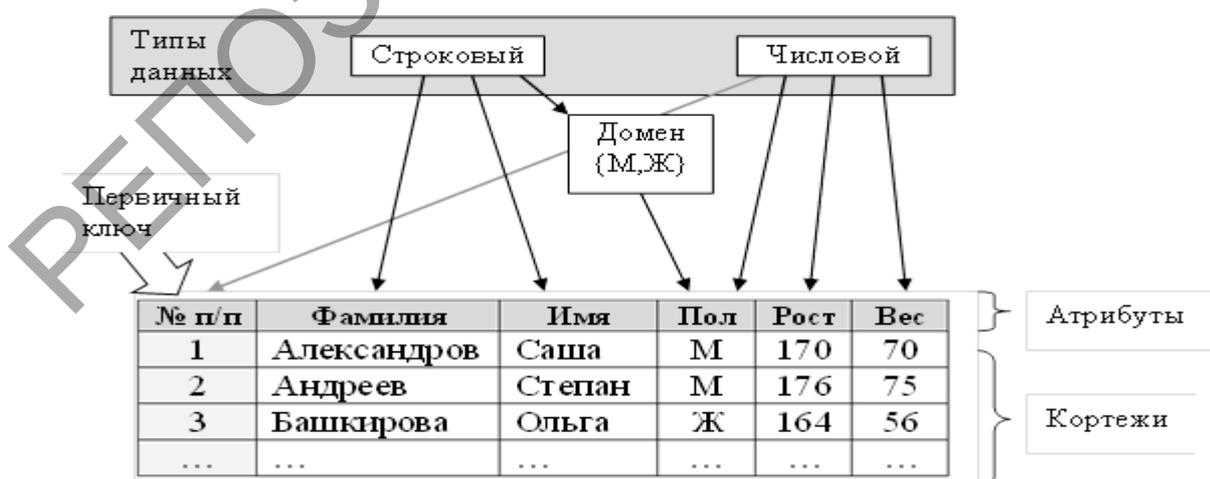


Рисунок 1 – Основные понятия РМД на примере фрагмента отношения «Студент»

## Лабораторная работа № 1. Создание базы данных «Библиотека» средствами системы управления базами данных Microsoft Access

### Задание:

1. Создать таблицы «Издательства», «Книги» и «Темы» базы данных «Библиотека».

### Методика выполнения:

1. Запустить Microsoft Access (версии 2008 или 2010).

2. Создать файл пустой базы данных «Библиотека».

3. Создать структуру таблицы «Издательства», которая содержит следующие поля: Код издательства, Наименование, Город (таблица 3). Имена полей таблиц из базы данных Библиотека: «Издательства», «Книги» и «Темы», типы данных, свойства полей, отличные от принципа умолчания, а также поля, являющиеся ключами, приведены в таблице 3.

Таблица 3. Структура таблиц базы данных «Библиотека»

Название таблицы	Имя поля	Тип данных	Свойства поля	Ключ
Издательства	Код издательства	Числовой	Размер поля – целое, обязательное поле	Да
Издательства	Наименование	Текстовый	Размер поля – 15	
Издательства	Город	Текстовый	Размер поля – 15	
Книги	Код книги	Числовой	Размер поля – целое, обязательное поле	Да
Книги	Название	Текстовый	Размер поля – 25	
Книги	Автор	Текстовый	Размер поля – 15	
Книги	Код издательства	Числовой	Размер поля – целое, обязательное поле	
Книги	Объем	Числовой	Размер поля – целое	
Книги	Год издания	Числовой	Размер поля – целое	
Книги	Стоимость	Денежный	Формат поля – денежный	
Темы	Код книги	Числовой	Размер поля – целое, индексированное поле (совпадения допускаются)	
Темы	Тема	Текстовый	Размер поля – 30	

4. После набора имен полей таблицы «Издательства», указания для них типы данных, установления требуемых свойств и определения ключа, нажать

на кнопку «Заккрыть» окна таблицы в режиме «Конструктор». На появившееся при этом сообщение:«Сохранить изменения макета или структуры таблицы «Таблица1»? ответить утвердительно. После этого в диалоговом окне«Сохранение» набрать имя таблицы «Издательства» и нажать копку «ОК».

5. Описанные выше действия выполнить для создания таблиц «Книги» и «Темы» базы данных «Библиотека».

6. Установить связи между таблицами «Издательства», «Книги» и «Темы» базы данных «Библиотека» так, как это показано на рисунке 2. Обратите внимание на то, что между таблицами базы данных «Библиотека» будут связи одного типа – «один ко многим».

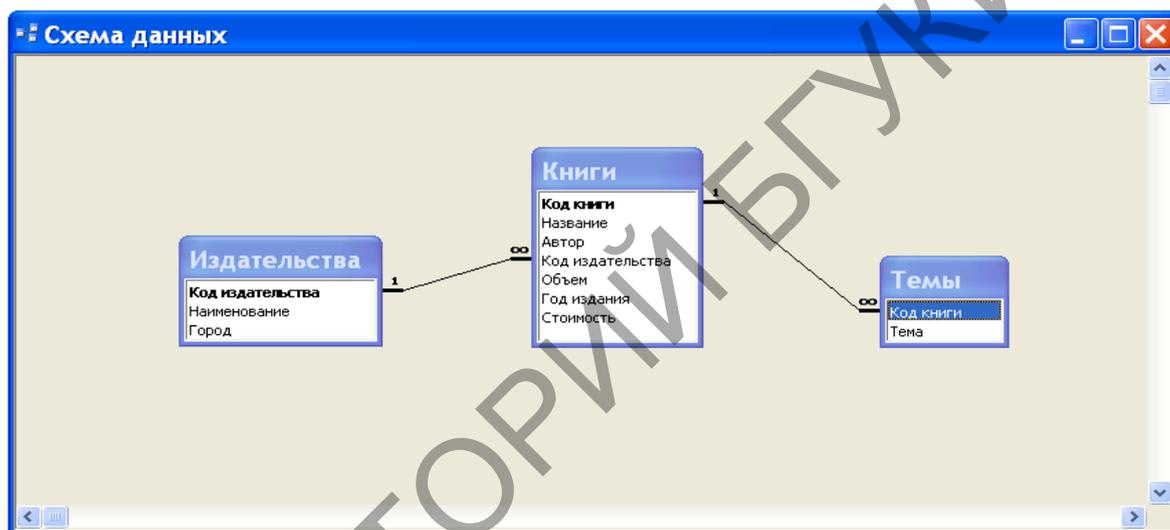


Рисунок 2 – Схема базы данных «Библиотека»

7. Ввести в таблицы «Издательства», «Книги» и «Темы» базы данных «Библиотека» данные, приведенные ниже в таблицах 4–6. Ввод данных в таблицу осуществляется в режиме таблицы. Чтобы открыть таблицу в режиме таблицы достаточно в окне базы данных во вкладке «Таблицы» выполнить двойной щелчок мышью на имени таблицы. Быстро перевести таблицу из режима «Конструктор» в режим «Таблица» и наоборот можно нажатием кнопки «Вид» на панели «База данных». Для перехода от одного поля к другому, когда осуществляется ввод данных в таблицу, удобно использовать клавишу «Tab».

Таблица 4. Издательства

Код издательства	Наименование	Город
1	Наука	Москва

2	Мир	Москва
3	Радио и связь	Минск
4	Машиностроение	Киев
5	Финансы и статистика	Москва

Таблица 5. Книги

Код книги	Название	Автор	Код издательства	Объем	Год издания	Стоимость
1	Педагогика	Беспалько	2	340	1994	24,00р.
2	Сборник задач	Сканави	2	634	1992	60,00р.
3	Программирование	Арсак	1	273	1989	18,00р.
4	Язык Ада	Перминов	3	278	1987	16,00р.
5	Операционные системы	Грибанов	3	446	1991	23,00р.
6	БД на Паскале	Ульман	4	563	1992	32,00р.
7	IBM PC для пользователя	Фигурнов	5	368	1994	22,00р.

Таблица 6. Темы

Код книги	Тема	Код книги	Тема
1	Личность человека	5	Управление заданиями
1	Проектирование ИПС	5	Управление задачами
1	Технология обучения	5	Управление данными
1	Анализ учебного процесса	6	Операции с поставщиками
2	Уравнения	6	Бухгалтерская книга
2	Прогрессии	6	Платежная ведомость
2	Геометрические задачи	6	Реляционная алгебра
3	Игры с числами	6	Правила нормализации
3	Игры без стратегии	7	Устройства компьютера
3	Комбинаторные задачи	7	Файлы и каталоги
3	Стратегия без игры	7	Диалог пользователя
4	Программные модули	7	Работа с дисками
4	Лексика	7	Программы архивации
4	Предопределенные типы	7	Конфигурирование системы
4	Операторы	7	Обслуживание дисков
5	Структура ОС ЕС	7	Редактирование текстов

Литература:

1. Как создать базу данных MicrosoftAccess за 30 минут[Электронный ресурс]. – Режим доступа :<https://www.youtube.com/watch?v=EQBjZmxcXsI>. – Дата доступа : 16.03.2020.

2. Как создать таблицы в MicrosoftAccess за 8 минут[Электронный ресурс]. – Режим доступа :[https://www.youtube.com/watch?v=h\\_UPuVGwkko](https://www.youtube.com/watch?v=h_UPuVGwkko). – Дата доступа : 16.03.2020.

## **Лабораторная работа № 2. Создание базы данных «Библиотека» средствами системы управления базами данных MicrosoftAccess**

### **Задание:**

1. Создать и добавить таблицы «Читатели» и «Выдача книг» в базу данных «Библиотека».

### **Методика выполнения:**

1. Открыть базу данных «Библиотека».

2. Создать в базе данных «Библиотека» структуру таблицы «Читатели», которая должна содержать следующие поля: «Код читателя», «Фамилия», «Имя», «Отчество», «Домашний телефон», «Домашний адрес». Типы данных для полей таблицы, их свойства определить самостоятельно по смыслу. В качестве ключа указать поле «Код читателя».

3. Создать в базе данных «Библиотека» структуру таблицы «Выдача книг», которая должна содержать следующие поля: «Код читателя», «Код книги», «Дата заказа». В этой таблице не задавать ключевое поле. Для поля «Дата заказа» указать тип данных – «Дата/время». Следует обратить внимание на то, что в последствии ключ «Код читателя» в таблице «Читатели» будет связываться с полем «Код читателя» в таблице «Выдача книг». Поэтому эти поля должны иметь соответствующие типы данных и свойства.

4. Установить между добавленными таблицами «Читатели» и «Выдача книг», а также созданными на лабораторном занятии № 1 таблицами «Издательства», «Книги» и «Темы», связи так, как это показано в окне «Схема данных» на рисунке 3.

Для установления связи между таблицами надо открыть окно «Схема данных». При его открытии появляется диалоговое окно «Добавление таблицы», в котором надо выделить имена тех таблиц, между которыми будут устанавливаться связи. После этого нажать кнопки «Добавить» и «Закрыть». Затем в окне «Схема данных» с помощью мыши переместить ключевое поле одной таблицы на соответствующее поле в другой таблице. В появившемся окне «Связи» задать режим «Обеспечение целостности

данных» и его подрежимы: «Каскадное обновление связанных полей» и «Каскадное удаление связанных записей» и нажать кнопку «Создать».

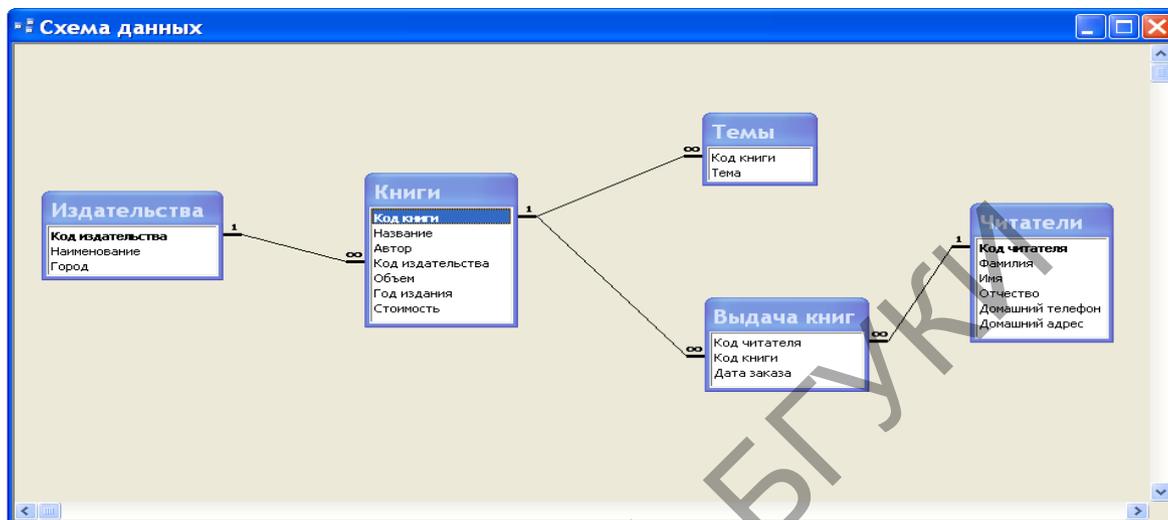


Рисунок 3 – Схема расширенной базы данных «Библиотека»

5. Открыть таблицу «Читатели» и ввести в нее данные, приведенные в таблице 7.

Таблица 7. Читатели

Код читателя	Фамилия	Имя	Отчество	Домашний телефон	Домашний адрес
1	Аксенов	Виктор	Сергеевич	252-88-13	ул. Есенина, 15-19
2	Голубева	Елена	Андреевна	220-99-29	ул. Чкалова, 7-38
3	Васильев	Игорь	Петрович	232-64-78	ул. Богдановича, 102-34
4	Кучеров	Валентин	Степанович	266-24-95	ул. Кнорина, 27-5
5	Мастяница	Вячеслав	Иванович	246-42-25	ул. Плеханова, 34-98
6	Победимская	Лариса	Анатолевна		ул. Чкалова, 9-10
7	Литвин	Борис	Николаевич	239-55-76	пр. Независимости, 46-54
8	Германович	Рита	Мироновна	278-31-51	ул. Казинца, 26-9
9	Бинцаровский	Теодор	Петрович		ул. Корженевского, 1-288

6. Открыть таблицу «Выдача книг» и ввести в нее данные, приведенные в таблице 8.

Таблица 8. Выдача книг

Код читателя	Код книги	Дата заказа	Код читателя	Код книги	Дата заказа
1	1	1.09.07	4	3	7.01.08
1	3	5.07.08	4	4	25.10.07
1	4	21.10.07	5	2	23.04.08
2	1	4.11.07	6	1	18.06.08
3	2	3.08.08	7	3	20.01.08
8	7	25.12.07	9	6	2.02.08

Литература:

1. Как создать базу данных MicrosoftAccess за 30 минут[Электронный ресурс]. – Режим доступа : <https://www.youtube.com/watch?v=EQBjZmxcXsI>. – Дата доступа : 16.03.2020.

2. Как создать таблицы в MicrosoftAccess за 8 минут[Электронный ресурс]. – Режим доступа : [https://www.youtube.com/watch?v=h\\_UPuVGwkko](https://www.youtube.com/watch?v=h_UPuVGwkko). – Дата доступа : 16.03.2020.

**Лабораторная работа № 3. Создание базы данных «Библиотека» средствами системы управления базами данных MicrosoftAccess**

**Задание:**

1. Создать простые запросы для выбора данных из базы данных «Библиотека».

**Методика выполнения:**

1. Открыть базу данных «Библиотека».

2. В режиме «Конструктор» создать запрос на выведение списка книг, цена которых находится в диапазоне от 20 до 30 рублей. Динамический набор этого запроса должен содержать поля: «Автор», «Название», «Год издания», «Стоимость». Для задания условия отбора вначале использовать оператор Between ... And, а затем операторы >=, <=, And. Записи в динамическом наборе расположить по возрастанию цены книг. Сохранить

первый запрос под именем «Цена книг из диапазона», а второй – под именем «Операторы сравнения для поиска цены».

3. Создать запрос на выведение списочитателей, у которых нет домашнего телефона. В список включить следующие поля: «Фамилия», «Имя», «Отчество», «Домашний адрес». Список рассортировать в алфавитном порядке по фамилии, имени и отчеству. Для поиска требуемых записей в строке «Условие отбора» для поля «Домашний телефон» использовать выражение IsNull. Это выражение предназначено для поиска записей, у которых поле не содержит значение (является пустым). Если требуется отобрать записи, у которых поле имеет значение, то можно использовать выражение IsNotNull. Запрос сохранить под именем «Читатели без домашних телефонов».

4. Создать запрос на выведение списочитателей, у которых в домашнем телефоне вторая цифра есть 5 или 6. В динамический набор включить поля: «Фамилия», «Имя», «Отчество», «Домашний телефон». Условие отбора для поля может иметь следующий вид: Like "[56]\*". Запрос сохранить под именем «Использование символов шаблона». Изменить условие отбора предыдущего запроса так, чтобы он выводил список всех читателей, в номерах телефонов которых вторая цифра не 5 и не 6. Полученный запрос сохранить под именем «Символ отрицания в квадратных скобках».

5. Создать запрос на выведение списка книг, заказанных читателями в 2007 году. В динамический набор включить следующие поля: «Автор», «Название», «Наименование», «Город». Для решения задачи вначале следует использовать функцию DatePart(interval; date; firstweekday; firstweek), а затем Format(expr; fmt; firstweekday; firstweek). Запросы сохранить под именами «Использование функцииDatePart» и «Использование функцииFormat» соответственно.

6. В таблицу «Выдача книг» базы данных «Библиотека» добавить поле «Дата возврата». В это поле для записей, приведенных в таблице 9, ввести даты возврата. В остальных записях поле «Дата возврата» должно остаться пустым.

Таблица 9. Учет возврата книг

Код читателя	Код книги	Дата заказа	Дата возврата
1	1	01.09.2007	15.10.2007
1	3	05.07.2008	23.09.2008
4	3	07.01.2008	02.03.2008
5	2	23.04.2008	03.05.2008

7	3	20.01.2008	11.04.2008
9	6	02.02.2008	03.03.2008

7. Составить запрос на выведение списка читателей, которые не сдали своевременно книги (предполагается, что читатель может держать книгу на руках не более 100 дней). В динамический набор включить следующие поля: «Фамилия», «Имя», «Отчество», «Домашний телефон», «Автор», «Название», «Стоимость». Для решения задачи следует воспользоваться функцией Date(). Запрос сохранить под именем «Читатели, не сдавшие своевременно книги».

#### Литература

1. Как создать запросы в MicrosoftAccess за 10 минут[Электронный ресурс]. – Режим доступа :<https://www.youtube.com/watch?v=Gshb4YHhJG8>. – Дата доступа : 16.03.2020.

2. Функция DateAdd в запросах Access[Электронный ресурс]. – Режим доступа :<https://www.youtube.com/watch?v=DV9ss2BiP78>.

### **Лабораторная работа № 4. Создание базы данных «Библиотека» средствами системы управления базами данных MicrosoftAccess**

#### **Задание:**

1. Создать формы для представления данных в базе данных «Библиотека».

#### **Методика выполнения:**

1. Открыть базу данных «Библиотека».

2. Для запроса «Стоимость книг с учетом инфляции» создать автоформу в столбец. Для создания автоформ надо выполнить следующие действия: в окне базы данных во вкладке «Формы» выполнить щелчок мышью на команде «Создать», а затем в появившемся диалоговом окне «Новая форма» выбрать в качестве источника данных требуемую таблицу или запрос, далее выбрать соответствующую автоформу и нажать кнопку «ОК». Форму для задания назвать «Автоформа в столбец».

3. Создать табличную автоформу, приведенную на рисунке4, и дать ей название.

Читатели, не сдавшие своевременно книги								
	Фамилия	Имя	Отчество	Домашний тел	Автор	Название	Стоимость	Дата заказа
▶	Аксенов	Виктор	Сергеевич	252-88-13	Перминов	Язык Ада	16 000,00р.	21.10.2007
	Кучеров	Валентин	Степанович	266-24-95	Перминов	Язык Ада	16 000,00р.	25.10.2007
*								

Запись: 1 из 2

Рисунок 4 – Табличная автоформа

4. Создать форму, приведенную на рисунке 5. Самостоятельно выяснить, какую таблицу или какой запрос для этой формы надо взять в качестве источника данных. Определить, какой стиль выбран для создания этой формы. Созданную форму назвать «Ленточная автоформа». Перечислить недостатки, которые есть в форме.

Ленточная автоформа						
№ читателя	Фамилия	Имя	Отчество	Домашний телефон	Домашний адрес	
1	Аксенов	Виктор	Сергеевич	252-88-13	ул. Есенина, 15-19	
2	Голубева	Елена	Андреевна	220-99-29	ул. Чкалова, 7-38	
3	Васильев	Игорь	Петрович	232-64-78	ул. Богдановича, 102-34	
4	Кучеров	Валентин	Степанович	266-24-95	ул. Кюрина, 27-5	
5	Мастяница	Владислав	Иванович	246-42-25	ул. Плеханова, 34-98	
6	Победимская	Лариса	Анатольевна		ул. Чкалова, 9-10	
7	Литвин	Борис	Николаевич	239-55-76	пр. Независимости, 46-54	
8	Германович	Рита	Мироновна	278-31-51	ул. Казинца, 26-9	
9	Бикцаровский	Теодор	Петрович		ул. Корженевская, 1-288	

Запись: 1 из 9

Рисунок 5 – Ленточная автоформа

5. Представить числовую информацию о стоимости книг из базы данных «Библиотека» в графической форме путем создания диаграммы, приведенной на рисунке 6.



Рисунок 6 – Стоимость книг

Возможности MSAccess для построения и форматирования диаграмм являются очень ограниченными. Поэтому дополнительную обработку диаграммы можно выполнить с помощью приложения MSGraph. Вызвать его можно в режиме конструирования формы двойным щелчком мыши в области диаграммы.

### Лабораторная работа № 5. Создание базы данных «Библиотека» средствами системы управления базами данных MicrosoftAccess

#### Задание:

1. Создать отчеты для представления данных в базе данных «Библиотека».

#### Методика выполнения:

1. Открыть базу данных «Библиотека».  
 2. Создать отчет «Выдача книг» с помощью мастера отчетов. Для этого необходимо выполнить следующие действия:

– В окне базы данных раскрыть вкладку «Отчет» или выбрать команду «Отчет» в меню «Вид».

– Выполнить двойной щелчок мышью на команде «Создание отчета с помощью мастера». На экране появится диалоговое окно «Создание отчетов» (рисунок 7).

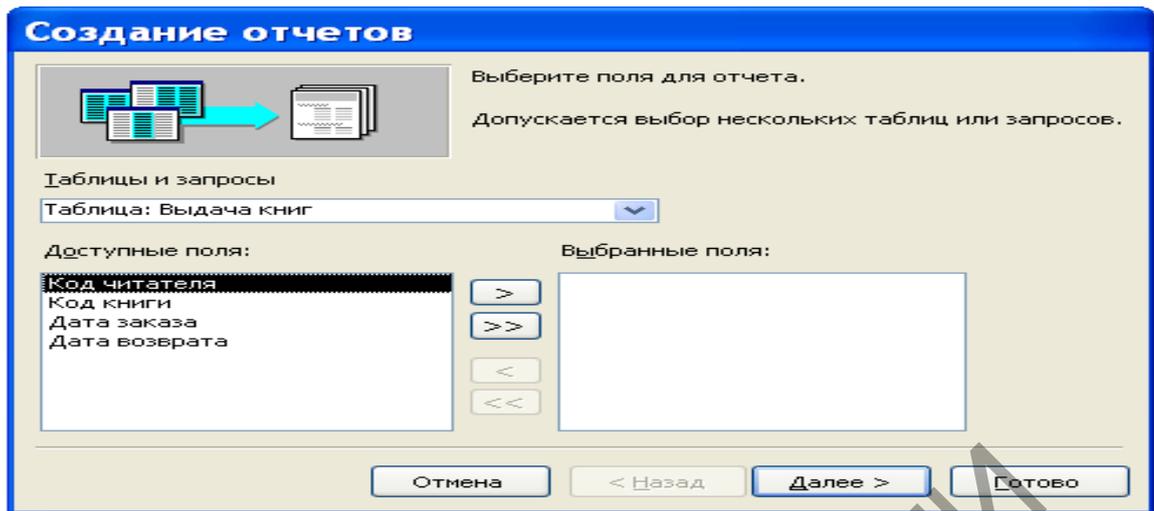


Рисунок 7 – Диалоговое окно «Создание отчетов»

– В этом окне из списка «Таблицы и запросы» выбрать требуемые имена, а затем в нижней части окна из левого столбца перенести требуемые поля в правый столбец, используя для переноса кнопки с символами >, >>. Кнопки <, << используются для отмены переноса полей. Выбирать можно несколько таблиц и запросов, при этом в той очередности, которая бы обеспечила требуемую очередность расположения полей в отчете.

– Нажимая кнопку «Далее», будет появляться очередное диалоговое окно, в котором надо действовать в соответствии с приведенными инструкциями. Нажатие кнопки «Готово» приведет к завершению построения отчета. Так, например, если в диалоговом окне, приведенном на рисунке 7, в качестве источника данных выбрать таблицу «Выдача книг», а затем с помощью кнопки >> перенести все поля из этой таблицы в отчет и нажать кнопку «Готово», то сразу получится отчет, приведенный на рисунке 8. Автоматически этому отчету будет присвоено имя «Выдача книг».

Код читателя	Код книги	Дата заказа	Дата возврата
1	4	21.10.2007	
	3	05.07.2008	23.09.2008
	1	01.09.2007	15.10.2007
2	1	04.11.2007	
	2	03.08.2008	
3	4	25.10.2007	
	3	01.07.2008	02.03.2008

Рисунок 8 – Отчет «Выдача книг»

3. Создать отчет, который будет для каждого читателя формировать

список прочитанных им книг. В этот список включить поля: «Фамилия», «Имя», «Отчество», «Автор», «Название» и «Стоимость». В качестве групповых итогов отчет должен для каждого читателя вычислять количество прочитанных книг и их суммарную стоимость. Суммарная стоимость книг должна иметь денежный формат. Примечание отчета должно содержать среднее количество прочитанных книг и среднюю стоимость книги. Отчет сохранить под именем «Рейтинг читателя».

4. Создать отчет, который будет вычислять для каждого издательства суммарное количество страниц книг. В отчет включить следующие поля: «Наименование из таблицы Издательства», а также «Название», «Автор», «Объем» из таблицы «Книги». Примечание отчета должно содержать информацию об общем количестве страниц книг. Дать имя отчету «Объем книг».

### **Лабораторная работа № 6. Создание базы данных «Библиотека» средствами системы управления базами данных Microsoft Access**

#### **Задание:**

1. Создать макросы для автоматизации работы приложения в базе данных «Библиотека».

#### **Методика выполнения:**

1. Открыть базу данных «Библиотека».

2. Создать макрос, который будет удалять встроенную панель инструментов режима формы. Для этой цели воспользоваться макрокомандой «ПанелиИнструментов», у которой аргументу «Панель инструментов» задать значение «режим формы», а аргументу «Показать» – значение «нет». Назвать созданный макрос «Удаление панели».

3. Создать макрос, приведенный на рисунке 9.

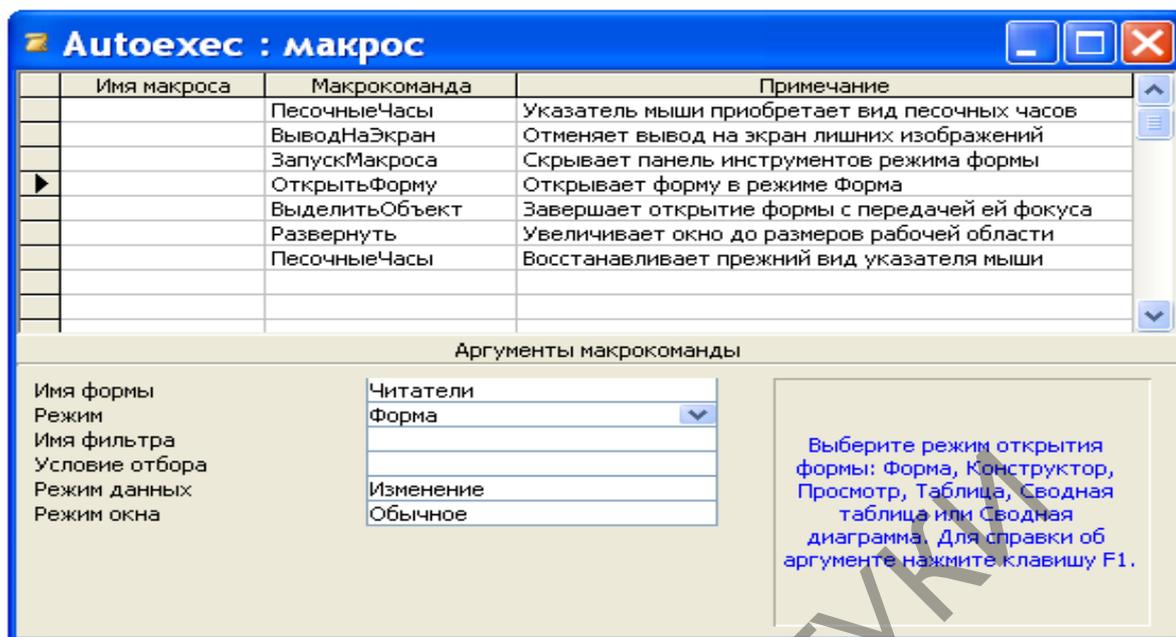


Рисунок9– Автоматически запускаемый макрос «Autoexec»

Следует обратить внимание, что этот макрос содержит семь макрокоманд, которые MS Access будет автоматически выполнять при открытии базы данных. Первая макрокоманда «ПесочныеЧасы» выводит указатель мыши в виде песочных часов, чтобы показать пользователю, что выполнение макроса может потребовать нескольких секунд. Следующая макрокоманда «ВыводНаЭкран» с аргументом «Включить вывод», установленным в значение «Нет», предназначена для того, чтобы на экран не выводились лишние изображения при выполнении макроса.

Макрокоманда «ЗапускМакроса» в данном случае должна запустить макрос «Удаление панели», который удалит встроенную панель инструментов режима формы.

Макрокоманда «ОткрытьФорму» открывает форму Читатели. На рисунке9 показано, что работу этой макрокоманды определяют четыре аргумента. Аргумент «Имя формы» предназначен для задания имя открываемой формы, а аргумент «Режим» устанавливает режим, в котором форма будет открыта. В примере используется режим «Форма».

По умолчанию для аргумента «Режим данных» устанавливается значение «Изменение», позволяющее пользователю добавлять, удалять и изменять записи во время работы с формой. Аргумент «Режим» окна по умолчанию принимает значение «Обычное». В этом случае форма открывается в режиме, заданном ее свойствами.

MS Access не всегда ждет завершения работы макрокоманды перед выполнением следующей. Можно завершить открытие формы, передав ей так называемый фокус. Это делается с помощью макрокоманды

«ВыделитьОбъект», в которой указывается объект, получивший фокус (в примере это форма Читатели). Затем макрокоманда «Развернуть» увеличивает активное окно (то есть окно, обладающее фокусом) до размеров рабочей области MS Access. Последняя макрокоманда «ПесочныеЧасы» в макросе «Autoexes» восстанавливает прежний вид указателя мыши, свидетельствуя об окончании работы макроса.

Макрос «Autoexes» всегда запускается автоматически при открытии базы данных, в которой он имеется. Если необходимо, чтобы этот макрос автоматически не запустился надо при открытии базы данных удерживать клавишу «Shift».

### **Лабораторная работа № 7. Создание базы данных «Библиотека» средствами системы управления базами данных MicrosoftAccess**

#### **Задание:**

1. Создать кнопочную форму для автоматизации работы приложения в базе данных «Библиотека».

#### **Методика выполнения:**

1. Открыть базу данных «Библиотека».

2. Создать главную кнопочную форму, которая похожа на форму, представленную на рисунке10. На этой форме имеются восемь кнопок: «Поступление книг», «Месячная загрузка», «Рейтинг книг», «Поиск книг», «Просмотр содержания», «Выдача книг», «Возврат книг», «Письменное уведомление», «Выход из Access», а также текст и рисунок.

В верхней части окна формы в режиме конструктора набрать текст, указанный на рисунке10. Для набора текста в форме используется кнопка «Надпись» на «Панели элементов». Нажать кнопку «Надпись» на «Панели элементов». Установить указатель мыши в области данных окна формы в режиме конструктора в том месте, где должен размещаться текст, и при нажатой левой клавише создать рамку надписи, а затем в ней набрать текст. Отформатировать набранный текст подходящим образом.

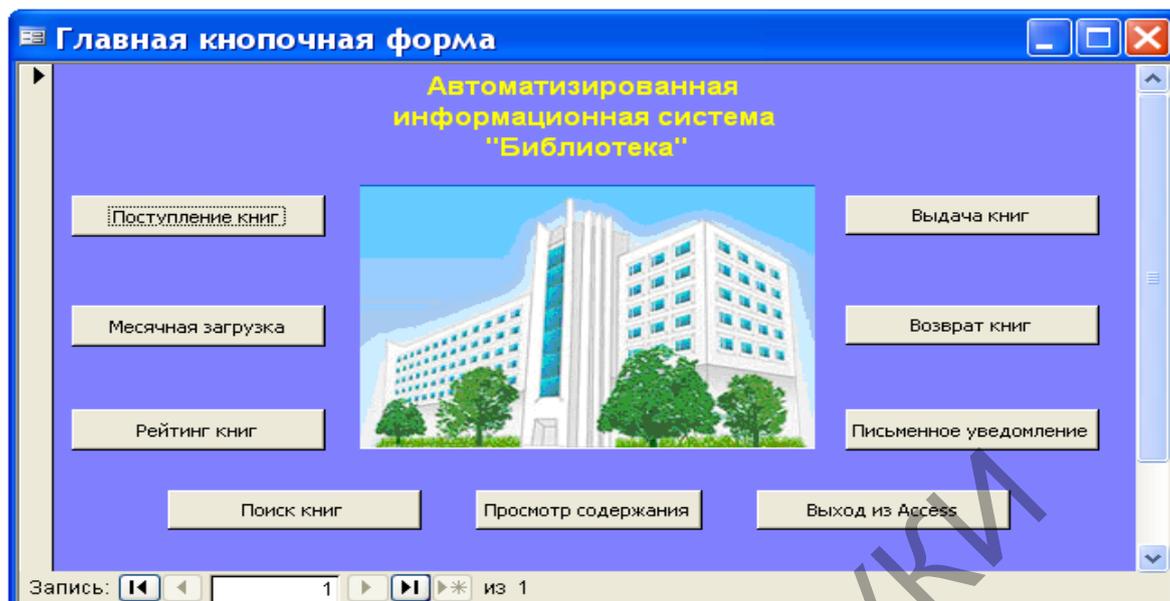
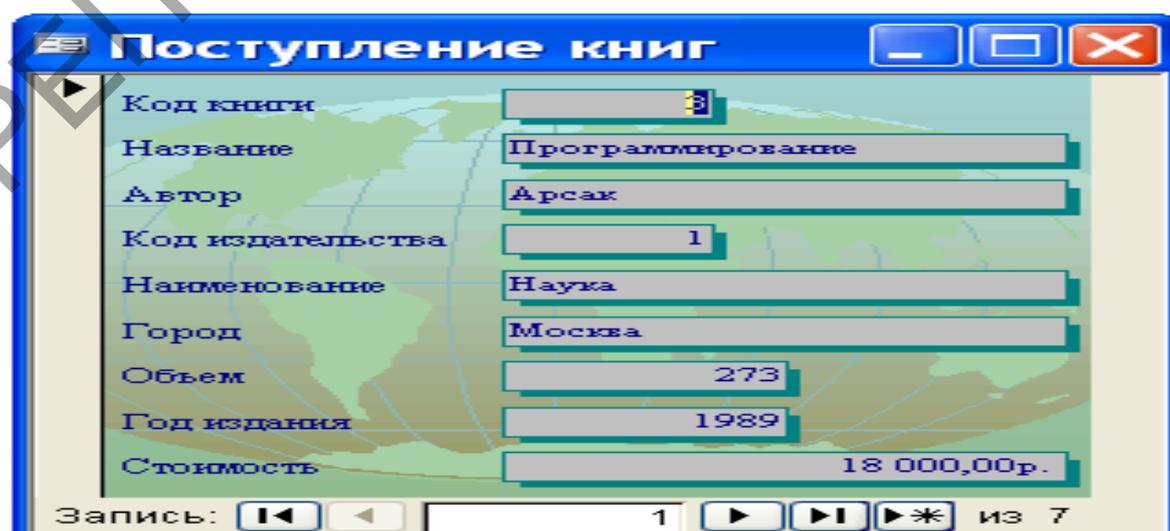


Рисунок10 – Главная кнопочная форма

3. Вместо фотографии с изображением БГУ культуры и искусств, размещенной в кнопочной форме (рисунок 10), вставить рисунок, который будет подходить для разрабатываемой информационной системы. Выбрать элемент «Рисунок» на «Панели элементов» и начертить прямоугольную область, в которую надо поместить рисунок. В диалоговом окне «Выбор рисунка» найти графический файл с требуемым рисунком и нажать кнопку «ОК».

4. Создать запрос, который будет содержать следующие поля: «Код книги», «Автор», «Название», «Код издательства», «Наименование», «Город», «Объем», «Год издания», «Стоимость». Запросу присвоить имя «Поступление книг». По данному запросу создать автоформу в столбец, которую также назвать «Поступление книг» (рисунок 11).



Выполнить щелчок мышью на элементе «Кнопка» на «Панели элементов» и изобразить прямоугольник в разделе «Область данных» главной кнопочной формы. В появившемся диалоговом окне мастера «Создание кнопок» выбрать категорию «Работа с формой» и действие «Открыть форму». В следующем диалоговом окне мастера кнопок указать в качестве источника данных форму «Поступление книг». В соответствующем диалоговом окне мастера кнопок указать, что на кнопке надо поместить текст «Поступление книг», и нажать на кнопку «Готово».

5. Для расчета количества выданных книг по месяцам ранее был составлен перекрестный запрос «Выдача книг по месяцам». Связать этот запрос с кнопкой «Месячная загрузка». Для этого в диалоговом окне мастера кнопок указать категорию «Разное», а действие – «Выполнить запрос». При указании имени источника данных в следующем диалоговом окне указать имя запроса «Выдача книг по месяцам».

6. Для создания кнопки «Поиск книг» воспользоваться ранее созданным запросом с параметром «Поиск книг по фамилии автора». Параметр позволяет набирать не все буквы фамилии автора, а только несколько первых букв. Диалоговое окно для ввода значения параметра в этом запросе имело вид, показанный на рисунке 12.

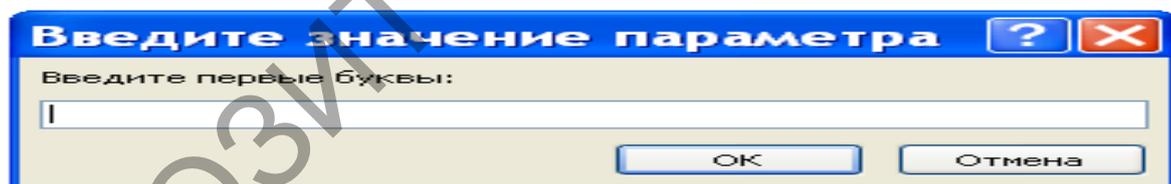


Рисунок 12– Диалоговое окно для ввода значения параметра

7. Чтобы можно было просматривать содержание книг, необходимо воспользоваться ранее созданной формой «Содержание книг». Для связывания кнопки «Просмотр содержания» с источником данных, являющимся формой «Содержание книг», в диалоговом окне мастера кнопок необходимо выбрать категорию «Работа с формой» и действие «Открыть форму», а в следующем диалоговом окне выбрать форму «Содержание книг», которая будет открываться нажатием данной кнопки.

8. Кнопка «Рейтинг книг» создается подобно кнопке «Просмотр содержания», но источником данных для нее будет не форма, а страница под названием «Books». Поэтому в диалоговом окне мастера кнопок необходимо

указать категорию «Работа с формой», а действие – «Открыть страницу». При указании имени страницы выбрать имя «Books».

9. Создать кнопку «Выдача книг», которая в качестве источника данных будет использовать форму «Выдача книг», содержащую поля: «Код читателя», «Код книги», «Дата возврата» и «Дата заказа».

10. При создании кнопки «Возврат книг» сначала создать запрос. Этот запрос должен для конкретного читателя выводить список книг, которые он не вернул в библиотеку. Для ввода информации о читателе создать параметр «Код читателя». В список книг включить следующие поля: «Фамилия», «Имя», «Отчество», «Автор», «Название», «Год издания», «Стоимость», «Дата заказа», «Дата возврата». Запрос назвать «Возврат книг».

После этого при создании кнопки «Возврат книги» в диалоговом окне мастера кнопок выбрать категорию «Разное», а в ней выбрать действие «Выполнить запрос». В следующем диалоговом окне в качестве источника данных указать созданный запрос «Возврат книг».

11. Кнопку «Письменное уведомление» создать точно таким же образом, как и предыдущую, но в качестве источника данных для нее необходимо использовать запрос с параметром «Список читателей для вызова». В качестве параметра в этом запросе задается количество дней, которые читатель может на руках держать книгу.

12. Кнопка «Выход из Access» предназначена для завершения работы с приложением MSAccess. Для ее создания необходимо выполнить следующее: в режиме конструктора форм с помощью мастера кнопок выбрать категорию «Приложение», а в ней – действие «Выйти из приложения».

## **Лабораторная работа № 8. Создание запросов с использованием языка реляционной алгебры**

### **Задание:**

1. Создать запросы, используя язык реляционной алгебры, на выборку данных.

### **Методика выполнения:**

#### **Вариант 1**

1. Рассмотреть следующие отношения, моделирующие работу банка и его филиалов:

– клиент может иметь несколько счетов, при этом они могут быть размещены как в одном, так и в разных филиалах банка;

– в отношении R1 содержится информация обо всех клиентах и их счетах в филиалах банка;

– каждый клиент, в соответствии со своим счетом, может рассчитывать на некоторый кредит от банка, сумма допустимого кредита также зафиксирована.

R1				
ФИО клиента	№ филиала	№ счета	Остаток	Кредит

R2	
№ филиала	Район

2. С использованием языка реляционной алгебры составить запросы, позволяющие выбрать:

- 1) Филиалы, клиенты которых имеют счета с остатком, превышающим \$1000.
- 2) Клиентов, которые имеют счета во всех филиалах данного банка.
- 3) Клиентов, которые имеют только по одному счету в разных филиалах банка. То есть в общем у этих клиентов может быть несколько счетов, но в одном филиале не более одного счета.
- 4) Клиенты, которые имеют счета в нескольких филиалах банка, расположенных только в одном районе.
- 5) Филиалы, которые не имеют ни одного клиента.
- 6) Филиалы, которые имеют клиентов с остатком на счету 0 (ноль).
- 7) Филиалы, у которых есть клиенты с кредитом, превышающим остаток на счету в 2 раза.

#### Вариант 2

1. Рассмотреть следующие отношения, моделирующие работу международной фирмы, имеющей несколько филиалов:

- 1) Филиалы фирмы могут быть расположены в разных странах, это отражено в отношении R1.
- 2) Клиенты фирмы также могут быть из разных стран, и это отражено в отношении R4.
- 3) По каждому конкретному заказу клиент мог заказать несколько разных товаров.

R1	
Филиал	Страна

R2		
Филиал	Заказчик	№ заказа

--	--	--

R3		
№ заказа	Товар	Количество

R4	
Заказчик	Страна

2. С использованием языка реляционной алгебры составить запросы, позволяющие выбрать:

- 1) Заказчиков, которые работают со всеми филиалами фирмы, но покупают только один товар.
- 2) Филиалы фирмы, которые торгуют всеми товарами.
- 3) Товары, которые фирма продает только в одной стране.
- 4) Заказчиков, которые работают с филиалами фирмы, которые расположены только в одной стране.
- 5) Филиалы, с которыми не работает ни один заказчик.
- 6) Заказчиков, которые работают только с филиалами, расположенными в той же стране, что и заказчик.
- 7) Заказчиков, которые покупают все товары, представленные в отношении R3.

### Вариант 3

1. Рассмотреть следующие отношения, моделирующие работу фирмы, занимающейся разработкой программных систем:

- 1) Каждый сотрудник административно закреплен только за одним отделом.
- 2) Файлы хранятся на разных серверах. На разных серверах файлы могут иметь одинаковые имена.
- 3) Создатель файла является его владельцем, поэтому у каждого файла только один владелец, но владелец файла может разрешить пользоваться файлом другим сотрудникам.
- 4) Существует множество системного программного обеспечения, каждая программа может работать с одним или с несколькими файлами, расположенными на одном или нескольких серверах.

R1	
Название файла	Имя владельца файла

R2		
Навание программы	Название файла	Сервер

R3	
Название файла	Название сервера

R4	
Сотрудник	Отдел

2.С использованием языка реляционной алгебры составить запросы, позволяющие выбрать:

- 1) Файлы, которые имеют нескольких пользователей из разных отделов.
- 2) Программы, которые работают только с одним файлом.
- 3) Файлы, которые имеют одно и тоже имя, но расположены на различных серверах и используются сотрудниками разных отделов.
- 4) Файлы, с которыми работают сотрудники всех отделов.
- 5) Файлы, пользователями которых являются сотрудники только одного отдела.
- 6) Программы, которые работают со всеми серверами.
- 7) Отделы, сотрудники которых не работают ни с одним файлом. То есть отделы, в которых нет ни одного сотрудника, работающего с каким-нибудь файлом.
- 8) Отделы, сотрудники которых работают со всеми серверами.
- 9) Серверы, с которыми работают сотрудники только одного отдела.

### Лабораторная работа № 9. Создание запросов на языке SQL

#### Задание:

1. Создать запросы на языке SQL

#### Методика выполнения:

##### Вариант 1

1. Рассмотреть ситуацию поступления в высшие учебные заведения, которая была характерна для периода, когда были разрешены так называемые репетиционные вступительные экзамены, которые сдавались раньше основных вступительных экзаменов в высшее учебное заведение.

Исходными являются три отношения R1, R2 и R3. Все они имеют эквивалентные схемы:

- R1= (ФИО, Паспорт, Школа);
- R2= (ФИО, Паспорт, Школа);
- R3= (ФИО, Паспорт, Школа).

Отношение R1 содержит список абитуриентов, сдававших репетиционные экзамены. Отношение R2 содержит список абитуриентов, сдававших экзамены на общих условиях. Отношение R3 содержит список абитуриентов, принятых в высшее учебное заведение. Считается, что при неудачной сдаче репетиционных экзаменов абитуриент мог делать вторую попытку и сдавать экзамены в общем потоке, поэтому некоторые абитуриенты могут присутствовать как в первом, так и во втором отношении.

2. Определить список абитуриентов, которые поступали два раза и не поступили в высшее учебное заведение.

3. Определить список абитуриентов, которые поступили в высшее учебное заведение с первого раза, то есть они сдавали экзамены только один раз и сдали их так хорошо, что сразу были зачислены в высшее учебное заведение.

4. Определить список абитуриентов, которые поступили в высшее учебное заведение только со второго раза.

5. Определить список абитуриентов, которые поступали только один раз в высшее учебное заведение и не поступили.

#### Вариант 2

1. Рассмотреть ситуацию сдачи экзамена студентами одной специальности.

Исходными являются три отношения R1, R2 и R3. Все они имеют эквивалентные схемы:

- R1= (ФИО, Номер\_зач);
- R2= (Дисциплина);
- R3= (Номер\_зач, Дисциплина).

Отношение R1 содержит список студентов, которые должны сдавать экзамены. Отношение R2 содержит список всех дисциплин, по которым студенты должны сдавать экзамены. Отношение R3 содержит список студентов, сдавших экзамены.

2. Определить список фамилий студентов, которые сдавали экзамены.

3. Определить список фамилий студентов, которые не сдали хотя бы один экзамен.

4. Определить список фамилий студентов, сдавших все экзамены.
5. Определить список фамилий студентов, сдавших все экзамены на 8 и 9.

### Вариант 3

1. Рассмотреть ситуацию сдачи сессии студентами некоторого учебного заведения.

Исходными являются три отношения R1, R2 и R3. Все они имеют эквивалентные схемы:

- R1= (ФИО, Дисциплина, Оценка);
- R2= (ФИО, Группа);
- R3= (Группы, Дисциплина).

R1 – информация о попытках (как успешных, так и неуспешных) сдачи экзаменов студентами. R2 – состав групп. R3 – список дисциплин, которые надо сдавать каждой группе.

2. Определить список фамилий студентов, которые сдали экзамен по дисциплине «Базы данных» на 9 баллов.

3. Определить список тех студентов, которые должны были сдавать экзамен по дисциплине «Базы данных», но пока еще не сдавали.

4. Определить список тех студентов, которые имеют несколько двоек.

5. Определить список отличников.

**Лабораторная работа № 10. Моделирование предметной области на этапе концептуального проектирования базы данных**

#### **Задание:**

1. Создать ER-модель предметной области.

#### **Методика выполнения:**

2. Ознакомиться с предметной областью, описанной в проекте.

### Вариант 1. Проект ПОСТАВКА ТОВАРОВ

Завод «Прогресс» поставляет товары (изделие А, изделие В, изделие С и др.) заказчикам по договорам. Для каждого товара определены планы поставок.

Необходимо спроектировать базу данных ПОСТАВКА ТОВАРОВ, информация которой будет использоваться для анализа выполнения заводом планов поставок.

В БД должна храниться информация:

– о ТОВАРАХ: код товара, наименование товара, цена товара (тыс. руб.);

– о ЗАКАЗАХ на поставку товаров: код заказа, наименование заказчика, адрес заказчика, телефон, номер договора, дата заключения договора, наименование товара, плановая поставка (шт.);

– о фактических ОТГРУЗКАХ товаров: код отгрузки, код заказа, дата отгрузки, отгружено товара (шт.).

При проектировании БД необходимо учитывать следующее:

– товар имеет несколько заказов на поставку. Заказ соответствует одному товару;

– товару могут соответствовать несколько отгрузок. В отгрузке могут участвовать несколько товаров.

Кроме того, следует учесть:

– товар не обязательно имеет заказ. Каждому заказу обязательно соответствует товар;

– товар не обязательно отгружается заказчику. Каждая отгрузка обязательно соответствует некоторому товару.

## Вариант 2. Проект РОЗНИЧНАЯ ТОРГОВЛЯ

Магазин розничной торговли продает персональные компьютеры, средства связи и периферийное оборудование: принтеры, накопители CD-RW и др.

Необходимо спроектировать базу данных РОЗНИЧНАЯ ТОРГОВЛЯ, информация которой будет использоваться для анализа продаж в магазине.

В БД должна храниться информация:

– о ТОВАРАХ: код товара, наименование товара, дата поступления в магазин, количество товара, цена закупки (руб.);

– о ПОСТАВЩИКАХ товаров: код поставщика, наименование поставщика, адрес, телефон, к кому обращаться;

– о ПРОДАЖАХ товаров в магазине: код продажи, код товара, дата продажи, количество проданного товара (шт.), цена розничная (руб.).

При проектировании БД необходимо учитывать следующее:

– поставщик поставяет несколько товаров. Товар поступает на склад магазина от нескольких поставщиков;

– товар имеет несколько продаж. Продажа относится к одному товару.

Кроме того, следует учесть:

– поставщик не обязательно поставяет товар (может временно не работать). Каждый товар обязательно поставяется;

– товар не обязательно продается. Каждая продажа обязательно связана с товаром.

### Вариант 3. Проект ТУРАГЕНСТВО

Работники турагентства продают путевки путешествий по разным странам. В каждую страну организуются несколько маршрутов. По каждому маршруту указывается цель путешествия (отдых, экскурсия, лечение, шоп-тур, обучение и др.).

Необходимо спроектировать базу данных ТУРАГЕНСТВО, информация которой позволит определять наиболее популярные маршруты за текущий год, отслеживать обращения клиентов и др.

В БД должна храниться информация:

- о СТРАНАХ: код страны, название страны, стоимость визы (руб.);
- о МАРШРУТАХ: код страны, код маршрута, наименование маршрута;
- о ПРОДАЖАХ: код маршрута, цель путешествия, цена путевки (руб.),

количество проданных путевок по маршруту, дата продажи.

При проектировании БД необходимо учитывать следующее:

- в каждую страну организуются несколько маршрутов. Маршрут имеет отношение только к одной стране;
- маршрут участвует в нескольких продажах. Продажа связана только с одним маршрутом.

Кроме того, следует учесть:

- по каждой стране обязательно организуется маршрут. Каждый маршрут обязательно имеет отношение к некоторой стране;
- маршрут не обязательно может участвовать в продаже (может быть невостребован). Каждая продажа *обязательно* связана с одним маршрутом.

### Вариант 4. Проект ПОДПИСНЫЕ ИЗДАНИЯ

Отделение Белпочты каждое полугодие осуществляет подписку граждан (в дальнейшем получателей) на различные издания (газеты, журналы) на один, три или шесть месяцев.

Необходимо спроектировать базу данных ПОДПИСНЫЕ ИЗДАНИЯ, информация которой будет использоваться для учета получателей и выписанных ими изданий.

В БД должна храниться информация:

– об ИЗДАНИЯХ, на которые можно оформить подписку: индекс издания, вид издания (газета, журнал), название издания, стоимость подписки на издание на 1 месяц (руб.);

– о ПОЛУЧАТЕЛЯХ: код получателя, Ф.И.О. получателя, адрес получателя (улица, дом, квартира);

– о ПОДПИСКАХ, осуществленных получателями: код получателя, индекс издания, срок подписки (в месяцах), месяц начала доставки издания, год начала доставки издания.

При проектировании БД необходимо учитывать следующее:

– получатель может осуществить подписку несколько раз (подписаться на несколько изданий). Каждая подписка осуществляется одним получателем;

– издание может быть использовано для нескольких подписок (на издание могут подписаться несколько получателей). Каждая подписка соответствует одному изданию.

Кроме того, следует учесть:

– каждый получатель обязательно осуществляет хотя бы одну подписку. Каждая подписка обязательно соответствует получателю;

– на издание не обязательно может быть подписка (оно может быть не востребовано). Подписка обязательно соответствует некоторому изданию.

3. Выделить сущности и обозначить их графически прямоугольниками. Каждую сущность описать набором атрибутов в виде схемы.

4. Выделить связи между сущностями и представить их графически ромбами.

5. Определить тип каждой связи и представить ее графически в виде ER-диаграммы.

6. Определить класс принадлежности каждой сущности и отразить его графически на ER-диаграмме связи.

7. Построить графическую ER-модель предметной области.

#### Литература

1. Сборник индивидуальных заданий по технологиям баз данных: учеб.-практ. пособие / М-во образования Респ. Беларусь, Бел. гос. экономич. ун-т ; В.С. Оскерко [и др.]. – Мн.: БГЭУ, 2005. – С. 6–10.

**Лабораторная работа № 11. Преобразование ER-модели предметной области в реляционную модель**

**Задание:**

1. Создать реляционную модель предметной области.

**Методика выполнения:**

1. Проанализировать созданную ER-модель предметной области в итоге выполнения лабораторной работы № 11.

2. Создать таблицу для каждой сущности и графически отобразить связи между таблицами на основе ER-диаграммы. При этом каждому атрибуту сущности должен соответствовать столбец таблицы. Генерация таблиц из ER-диаграмм должна осуществляться в соответствии с правилами, которые опираются на два основных фактора – тип связи и класс принадлежности.

3. Полученные таблицы связать посредством связи первичных и внешних ключей.

**Литература**

1. Сборник индивидуальных заданий по технологиям баз данных: учеб.-практ. пособие / М-во образования Респ. Беларусь, Бел. гос. экономич. ун-т ; В.С. Оскерко[и др.]. – Мн.: БГЭУ, 2005. – С. 11–15.

**Лабораторная работа № 12. Составление паспорта базы данных****Задание:**

1. Составить паспорта на базы данных следующих библиотек Республики Беларусь:

– Президентская библиотека Республики Беларусь;

– Центральная научная библиотека имени Якуба Коласа Национальной академии наук Беларуси;

– Белорусская сельскохозяйственная библиотека имени И.С. Лупиновича Национальной академии наук Беларуси.

**Методика выполнения:**

1. Запустить любой браузер сети Интернет.

2. Зайти на страницы библиотек, указанных в задании.

3. Провести анализ информации о базах данных, размещенной в библиотеках как в локальном, так и удаленном доступе на страницах в сети Интернет. Отобрать 6 баз данных, которые генерируются или приобретаются библиотеками, и составить на каждую из них паспорт по общей форме:

**ПАСПОРТ**

№ п/п	Графа	Содержание
1.	Название БД	

2.	Автоматизированная система БД	
3.	Версия БД	
4.	Дата создания БД	
5.	Владелец БД	
6.	Кем создается БД	
7.	Тип БД	
8.	Вид БД	
9.	Назначение БД	
10.	Тип документов (объектов)	
11.	Тематика БД	
12.	Тип документов (объектов)	
13.	Естественный язык (русский,...)	
14.	Число записей	
15.	Период обновления (ежедневно, еженедельно, ежемесячно, другое)	
16.	Интенсивность обновления (записей)	
17.	Режим обновления (добавление / замена / удаление записей)	
18.	Ежегодный прирост (записей)	
19.	Операционная система	
20.	Хронологические рамки	с _____ года по _____ год
21.	Информационно-поисковые языки (ББК, предметные рубрики, ключевые слова)	
22.	Другие поисковые возможности	
23.	Степень свертывания информации о документе (БО, реферат, аннотация, ключевое слово)	

Дата составления паспорта « \_\_\_\_ » \_\_\_\_\_ 20 г.

Положение (дата утверждения « \_\_\_\_ » \_\_\_\_\_ 20 г.

Ответственный за ведение БД (Ф.И.О., должность \_\_\_\_\_

В графе «Название БД» указывают полное наименование базы данных.

При заполнении графы «Автоматизированная система» указывают наименование автоматизированной системы, в составе которой функционирует описываемая база данных.

В графе «Версия БД» указывают номер версии базы данных в свободной форме.

При заполнении № 5–8, 10,11, 13–16, 18, 21 граф необходимо иметь в виду следующее:

5\*. Заносят сведения о юридическом лице, обладающем правами собственности на базу данных. Указывают полное наименование организации.

6\*. Перечисляют все отделы, принимающие непосредственное участие в организации, ведении и редактировании базы данных, в скобках указывают обязанности.

7\*. Указывают тип базы данных. Перечень значений с пояснениями приведен в приложении ГОСТа 7.70–96. «Описание баз данных и машиночитаемых информационных массивов. Состав и обозначение характеристик».

8\*. Указывают вид базы данных, например, текущая, законсервированная и т.п.

10\*. Для документальных баз данных указывают объекты, например: книги, статьи, брошюры, издания, неопубликованные документы, патентные документы и др. Для остальных типов баз данных допускается запись в виде свободного текста, например: объектами в различных видах фактографических баз данных могут являться события, свойства, компьютерные программы, планы, расписания, информация для населения, словари, тезаурусы, номенклатуры, природные явления, технологические процессы и т.д.

11\*. Раскрывают тематику базы данных: универсального содержания или это может быть область деятельности, отрасль народного хозяйства, научная дисциплина, к которым относится база данных.

13\*. Указывают число описанных в базе данных объектов, с указанием типа объекта (документов, статей и т.д.). Например, 102855 записей документов.

14\*. Указывают период времени между очередными актами пополнения (обновления) базы данных. Например: ежемесячно; ежедневно; по мере поступления новых документов и т.д.

15\*. Указывают объем изменения (пополнения) базы данных за какой-либо период.

Способ записи: абсолютное значение или процентное отношение к общему объему базы данных. Например: по количеству вновь поступивших документов; 200 записей в день; около 1300 записей в месяц и т.д.

16\*. Указывают характер изменений, вносимых при обновлении данных в соответствии со следующим перечнем: полная замена данных; добавление записей; удаление записей; замена в отдельных полях записи; сохранение прежних версий.

18\*. В свободной форме указывают полное или сокращенное наименование операционной системы, ее версию, граничные значения версии, диапазон версий.

21\*. Раскрывают другие поисковые возможности: автор; организация/мероприятие; год издания; персоналии; заглавие; штрих-код/инвентарный номер и др.

### **Лабораторная работа № 13. Создание средств защиты базы данных «Библиотека» инструментарием системы управления базами данных MicrosoftAccess**

#### **Задание:**

1. Установить пароль в базе данных «Библиотека».

#### **Методика выполнения:**

1. Открыть базу данных «Библиотека», созданную в результате выполнения лабораторных работ № 1–7.

2. Создать для базы данных «Библиотека» пароль и запомнить его. В противном случае невозможно будет ни открыть базу данных «Библиотека», ни удалить из нее пароль.

3. Удалить пароль из базы данных «Библиотека».

4. Создать собственную форму для проверки пароля входа в базу данных «Библиотека». Для этого выполнить следующее:

– На панели «Объекты» окна базы данных выполнить щелчок мышью на кнопке «Формы», а затем двойной щелчок мышью на команде «Создание формы в режиме конструктора».

– Выполнить щелчок правой клавишей мыши в разделе данных формы и в появившемся контекстном меню выбрать команду «Свойства». После этого в диалоговом окне «Раздел: ОбластьДанных» задать высоту раздела данных 5 см. Закрыть это диалоговое окно.

– Выполнить двойной щелчок мышью на маркере выделения формы (он расположен в левом верхнем угле окна формы на пересечении вертикальной

и горизонтальной линеек). В появившемся окне свойств формы отключить вывод полос прокрутки, области выделения, кнопок перехода, разделительных линий, кнопки оконного меню, кнопок размеров окна, кнопки закрытия окна, задать ширину формы 8 см. Закрыть окно свойств формы.

– Выбрать элемент «Поле» на «Панели элементов» и с его помощью определить в центральной части области данных формы место для поля, в которое будет вводиться пароль.

– Выполнить щелчок правой клавишей мыши на данном поле. В контекстном меню выбрать команду «Свойства». В появившемся окне в строку «Имя» ввести имя «Вход». Закрыть окно свойств.

– Выбрать элемент «Надпись» на «Панели элементов» и сверху над полем «Вход» создать рамку для надписи. Ввести в нее текст «Введите пароль».

– Сохранить созданную форму под именем «Начало работы». Форма должна быть похожа на форму, приведенную на рисунке 13. Исключение составляет наличие в этой форме кнопки «ОК», которая будет создана позже. Закрыть форму «Начало работы».

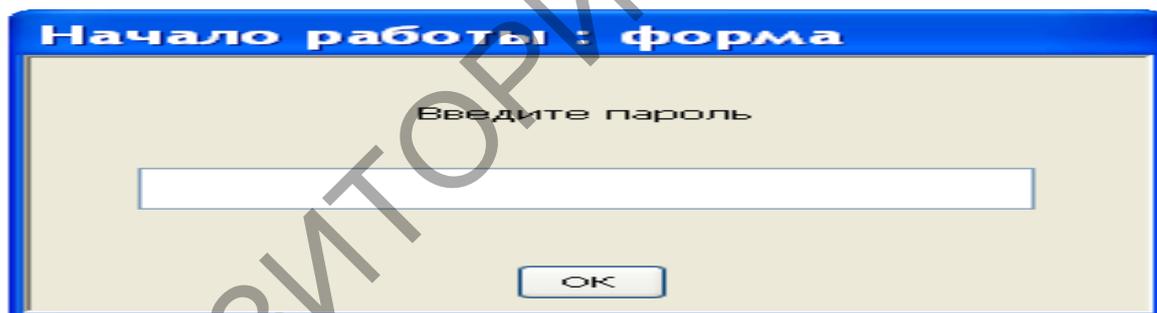


Рисунок 13 – Форма «Начало работы» в режиме формы

5. Создать макрос для проверки пароля входа в базу данных «Библиотека» (рисунок 14):

– Выполнить щелчок мышью на кнопке «Макросы» на панели «Объекты» окна базы данных. После этого нажмите кнопку «Создать» на панели инструментов окна базы данных.

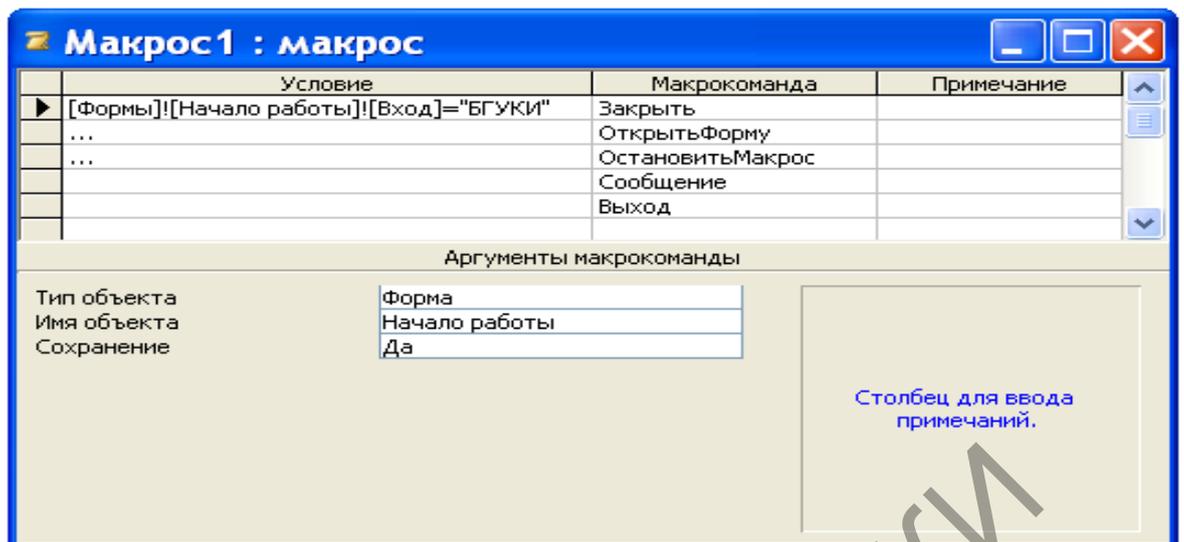


Рисунок 14 – Макрос «Защита» в режиме конструктора

– В окне конструктора макроса отобразить столбец условий, нажав мышью на кнопку «Условия», расположенную на панели «Конструктор макросов».

– В первую строку столбца «Условие» ввести следующее условие: [Формы]![Начало работы]![Вход]='БГУКИ'.

– Для первой строки столбца «Макрокоманда» выбрать из списка макрокоманду «Закреть». Задать для нее следующие аргументы: «Тип объекта» – «Форма», «Имя объекта» – «Начало работы»;

– Во вторую строку столбца «Условие» ввести многоточие (...), для этой же строки в столбец «Макрокоманда» из списка макрокоманд выбрать макрокоманду «ОткрытьФорму». В нижней части окна в качестве аргумента «Имя формы» ввести название формы «Библиотека».

– В третьей строке в столбец «Условие» ввести многоточие, а в столбец «Макрокоманда» – макрокоманду «ОстановитьМакрос».

– В четвертую и пятую строки ввести макрокоманды, которые надо выполнить в том случае, если пароль будет введен неверно. Для этого из списка макрокоманд для четвертой строки выбрать макрокоманду «Сообщение» и задать для нее следующие аргументы: «Сообщение» – «Постарайтесь вспомнить его», «Сигнал» – «Да», «Тип» – «Отсутствует», «Заголовок» – «Пароль неверный!». В пятую строку в столбец «Макрокоманда» выбрать из списка макрокоманду «Выход».

– Сохранить созданный макрос под именем «Защита» и закрыть его.

6. Создать кнопку «ОК» в форме «Начало работы», которая будет запускать макрос «Защита» для проверки пароля. Вид этой кнопки показан на рисунке 13. Выполнить следующие действия:

– Открыть форму «Начало работы» в режиме конструктора.

– Выполнить щелчок мышью на элементе «Кнопка» на «Панели элементов» и указать место в форме, где будет находиться кнопка, запускающая макрос для проверки пароля.

– Выполнить щелчок правой клавишей мыши на созданной кнопке и из контекстного меню выбрать команду «Свойства».

– В окне свойств кнопки выполнить следующее: для свойства «Имя» указать значение «ОК», для свойства «Подпись» указать также значение «ОК», для свойства «Нажатие кнопки» из списка выбрать имя макроса «Защита»

– Закрыть окно свойств кнопки и форму «Начало работы».

7. Создать макрос «Autoexes», показанный на рисунке 15. Макрос «Autoexes» обладает уникальным свойством – при запуске приложения он сразу же начинает выполняться. В этот макрос включить в указанной последовательности следующие макрокоманды: «ОткрытьФорму», «ОстановитьМакрос». Для макрокоманды «ОткрытьФорму» указать аргумент «Имя формы» – «Начало работы».

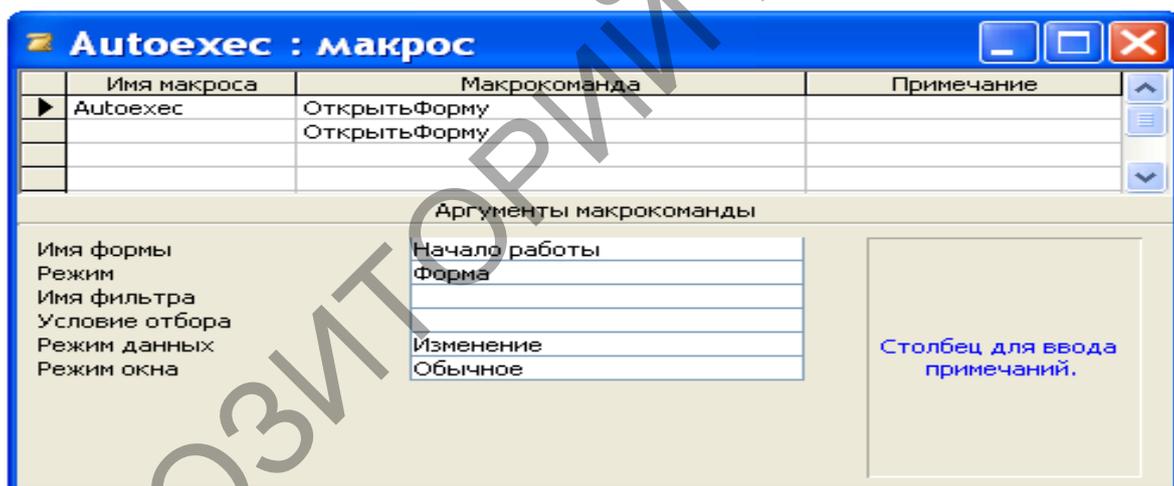


Рисунок15– Макрос в режиме конструктора

8. Закрыть базу данных «Библиотека». Выполнить запуск базы данных «Библиотека» при верном и неверном пароле.

#### **Лабораторная работа № 14. Выявление средств защиты баз данных собственной генерации в библиотеках Республики Беларусь**

##### **Задание:**

1. Определить наличие средств защиты баз данных собственной генерации в следующих библиотеках Республики Беларусь:

– Президентская библиотека Республики Беларусь;

- Центральная научная библиотека имени Якуба Коласа Национальной академии наук Беларуси;
- Белорусская сельскохозяйственная библиотека имени И.С. Лупиновича Национальной академии наук Беларуси;
- Республиканская научная медицинская библиотека;
- Республиканская научно-техническая библиотека.

**Методика выполнения:**

1. Посетить одну из библиотек, указанных в задании.
2. Ознакомиться с сегментом баз данных (кроме электронного каталога), генерируемых в библиотеке.
3. Отобрать 5 баз данных. Провести анализ условий и возможностей доступа и использования каждой базы данных, обращая внимание на наличие таких средств ее защиты, как: а) пароль; б) шифрование данных и программ; в) права доступа к объектам базы данных; г) защита полей и записей таблиц базы данных; д) встроенные средства контроля значений данных в соответствии с типами; е) блокировки. Указать другие средства защиты баз данных в случае их обнаружения.
4. Результаты анализа средств защиты баз данных отразить в таблице 10 «Средства защиты баз данных собственной генерации в библиотеках Республики Беларусь». В случае наличия в базе данных средств защиты, указанных в заголовках столбцов таблицы, в графах необходимо проставить знак «+». Названия других средств, используемых для защиты баз данных в библиотеке, привести в соответствующей графе.

Таблица 10. Средства защиты баз данных собственной генерации в библиотеках Республики Беларусь

Название библиотеки								
№ п/п	Название базы данных	Средства защиты базы данных						
		Пароль	Шифрование данных и программ	Права доступа к объектам базы данных	Защита полей и записей таблиц базы данных	Встроенные средства контроля значений данных	Блокировки	Другие средства

РЕПОЗИТОРИЙ БГУКИ

### **3.3 Методические указания к семинарским занятиям**

Цель семинарских занятий – научить студентов самостоятельно анализировать учебную и научную литературу и вырабатывать у них опыт самостоятельного мышления по проблемам учебной дисциплины.

Семинарские занятия посвящаются ключевым темам учебной дисциплины и проводятся в форме совместного обсуждения определенных вопросов по темам. Подготовка к семинару требует изучения и конспектирования основной и дополнительной литературы с целью изложения полных ответов на поставленные вопросы. В ответах обязательно должны приводиться фамилии авторов, подходы и мнения которых отмечаются и анализируются в них.

В процессе семинарских занятий студент должен:

- закрепить знания, усвоенные на лекциях и в процессе самостоятельной работы;
- организовать самоконтроль по усвоению основных теоретических положений, фактов, понятий, терминов, имен ученых и специалистов;
- развить навыки анализа литературы, участия в дискуссиях, выступлениях с докладами, рефератами, сообщениями.

По итогам семинарского занятия студенту проставляется оценка с учетом его самостоятельных ответов и дополнений по обсуждаемым вопросам. В случае отсутствия студента на семинарском занятии, его тема должна быть отработана по согласованию с преподавателем

### **3.4 Тематика семинарских занятий**

## **Семинарское занятие № 1. Теоретические и технико-технологические вопросы создания и функционирования распределенных баз данных**

### *Вопросы:*

1. Понятия «система распределенных баз данных», «распределенная база данных», «система управления распределенной базой данных». Принципы создания и функционирования распределенных баз данных.
2. 12 свойств (правил) распределенных баз данных, сформулированные К. Дж. Дейтом.
3. Модели распределения информации: архитектура файл-сервер, архитектура клиент-сервер, многоуровневая архитектура, Интернет/Интранет архитектура.
4. Однородные и неоднородные системы построения распределенных баз данных. Методы построения распределенных баз данных «сверху вниз» и «снизу вверх».
5. Характеристики «идеальной» распределенной системы управления базами данных. Четырехуровневая схема доступа к распределенным данным: «Удаленный запрос», «Удаленная транзакция», «Распределенная транзакция», «Распределенный запрос».
6. Преимущества и недостатки распределенных систем управления базами данных. Архитектура распределенного хранилища данных.

### Литература

1. Агальцов, В.П. Базы данных [Текст]: учебник. В 2 кн. Кн. 2. Распределенные и удаленные базы данных / В.П. Агальцов. – Москва : ФОРУМ : ИНФРА-М, 2017. – 272 с.
2. Аносова, Н.П. Распределенные базы и хранилища данных [Электронный ресурс] / Н.П. Аносова, О.О. Бородин, Е.С. Гаврилов, А.М. Марасанов ; Национальный Открытый Университет «ИНТУИТ». – Режим доступа : <https://www.intuit.ru/studies/courses/1145/214/info>.
3. Бобрешов-Шишов, Д.И. Динамическое управление структурой распределенной базы данных [Электронный ресурс] / Д. И. Бобрешов-Шишов, Л.А. Саяркин, И.А. Шаров // Молодой ученый. – 2015. – № 7 (87). – С. 51–53. – Режим доступа : <https://moluch.ru/archive/87/16944/>.
4. Гущин, А.Н. Распределенные базы данных и хранилища данных

[Электронный ресурс]/ А.Н. Гущин // Базы данных: учебно-методическое пособие / А.Н. Гущин. – 2-е изд., испр. и доп. – Москва; Берлин : Директ-Медиа, 2015. – С. 173–184. – Режим доступа : [http://biblioclub.ru/index.php?page=book\\_view\\_red&book\\_id=278093](http://biblioclub.ru/index.php?page=book_view_red&book_id=278093). – Дата доступа : 6.05.2020.).

5. Дейт, К.Дж. Распределенные базы данных [Текст]/ К.Дж. Дейт // Введение в системы баз данных : пер. с англ. / К.Дж. Дейт. – 8-е изд. – Москва; Санкт-Петербург; Киев : Вильямс, 2005. – Гл. 21. – С. 821–869.

6. Распределенные базы данных [Электронный ресурс]// Управление данными : учебник / Ю.Ю. Громов, О.Г. Иванова, А.В. Яковлев, В.Г. Однолько ; Тамбовский государственный технический университет. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2015. – Гл. 5, § 5.2. – С. 94–102. – Режим доступа : <http://biblioclub.ru/index.php?page=book&id=444642>. – Дата доступа : 06.05.2020.

7. Распределенные базы данных [Электронный ресурс]: учебное пособие / авт.-сост. Н.Ю. Братченко ; Северо-Кавказский федеральный университет. – Ставрополь : СКФУ, 2015. – 130 с. – Режим доступа : <http://biblioclub.ru/index.php?page=book&id=457594>. – Дата доступа : 06.05.2020.

8. Советов, Б.Я. Базы данных [Текст] : учебник для прикладного бакалавриата / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. – 3-е изд., перераб. и доп. – М. : Юрайт, 2018. – С. 199–249.

9. Чертовской, В.Д. Базы и банки данных [Электронный ресурс]: учеб. пособие / В.Д. Чертовской ; Моск. гос. ун-т печати. – СПб: [МГУП], 2001. 220 с. – Режим доступа : <http://www.hi-edu.ru/e-books/xbook099/01/>.

## **Семинарское занятие № 2. Практический опыт формирования и использования баз данных в библиотеках**

### *Вопросы:*

1. Виды и состав баз данных, приобретаемых и генерируемых библиотеками учреждений высшего образования. Организационно-технологические и технические особенности генерации баз данных.

2. Виды и состав баз данных, приобретаемых и генерируемых библиотеками научно-исследовательских учреждений, научных обществ и

академий. Организационно-технологические и технические особенности генерации баз данных.

3. Виды и состав баз данных, приобретаемых и генерируемых национальными библиотеками. Организационно-технологические и технические особенности генерации баз данных.

4. Модели приобретения баз данных библиотеками и обеспечения к ним доступа в современных условиях функционирования и развития информационно-документного рынка.

## Литература

### Основная

1. Анохин, А.А. Развитие электронных информационных ресурсов библиотеки учреждения высшего образования в оценке их востребованности пользователями [Электронный ресурс] / А.А. Анохин // Веснік Беларускага дзяржаўнага ўніверсітэта культуры і мастацтваў. – 2017. – № 1. – С. 146–155. – Режим доступа : <http://repository.buk.by/bitstream/handle/123456789/12979/Razvitie%20elektronnih%20informacionnih%20resursov%20biblioteki%20uchrejdeniya%20visshego%20obrazovaniya%20v%20ocenke%20ih%20vostrebovannosti%20polzovatelyami.pdf?sequence=1&isAllowed=y>.

2. Арлоўская, Т.Л. Паўнатэкставыя базы дадзеных краязнаўчага месцу: фарміраванне і выкарыстанне ў бібліятэцы [Тэкст] / Т.Л. Арлоўская // Бібліяпанарама : БП : інфармацыйна-аналітычнае выданне / УК «Абласная бібліятэка імя М. Горкага», Аддзел бібліятэказнаўства. – 2013. – Вып. 17. – С. 72–74.

3. Кузьминич, Т. Корпоративность как приоритет современной информационно-ресурсной политики НББ [Текст] / Т. Кузьминич // Интерактивные центры науки и технологий – новые возможности развития общества : м-лы науч.-практ. междунар. конф., посвящ. открытию первого в Респ. Беларусь интерактивного центра науки и технологий – Мультицентра «Совушка», Гродно, 17–18 декабря 2015 года / Гродн. обл. науч. б-ка ; [редкол. : В.Н. Жук, Л.С. Лотыш, Л.В. Мальцева]. – Гродно, 2015. – С. 37–53

4. Кузьминич, Т.В. Фактографические базы данных Национальной библиотеки Беларуси — источник информации о национальных объектах [Текст] / Т. В. Кузьминич // Бібліятэчны веснік : [навуковы зборнік

артыкулаў] / Нацыянальная бібліятэка Беларусі. – Мінск. – 2010. – Вып. 2. – С. 158–166.

5. Лихимович, О.В. База данных «Беларусь в лицах и событиях» – информационный ресурс национального уровня [Текст] / О.В. Лихимович, Е.Н. Малиновская // Адукацыя і выхаванне. – 2012. – № 1. – С. 68–72.

6. Мотульский, Р.С. Доступ к национальным и мировым электронным информационным ресурсам в Национальной библиотеке Беларуси [Текст] / Р.С. Мотульский // Infolib : информационно-библиотечный вестник. — 2015. – № 1. – С. 20–22.

7. Понтаплева, Е.С. Методика создания фактографических баз данных [Электронный ресурс] / Понтаплева Е.С. // Информационный бюллетень РНТБ / Государственный комитет по науке и технологиям Республики Беларусь, РНТБ. – 2014. – № 2. – С. 56–58. – Режим доступа : [http://rntbcat.org.by/Cont/bul/ibrlist\\_2\\_2014.pdf](http://rntbcat.org.by/Cont/bul/ibrlist_2_2014.pdf).

8. Предеина, А.В. Базы данных библиотек: актуальные вопросы создания и использования [Текст] / А.В. Предеина // Бібліяпанарама : БП : інфармацыйна-аналітычнае выданне / УК «Абласная бібліятэка імя М. Горкага», Аддзел бібліятэказнаўства. – 2019. – Вып. 22. – С. 102–106.

9. Соболевская, О. Виртуальный читальный зал Национальной библиотеки Беларуси как эффективный инструмент формирования информационных ресурсов публичных библиотек [Текст] / О. Соболевская // Бібліятэчны свет. – 2017. – № 3. – С. 25–28.

10. Сухотина, М.Л. Создание ресурсов по культуре и искусству на русском языке национальными библиотеками СНГ [Текст] / М.Л. Сухотина // Библиотековедение. – 2018. – Т. 67, № 6. – С. 617–629.

#### *Дополнительная*

1. Бричковский, В.И. Интеграция краеведческих электронных информационных ресурсов Беларуси на сайте «Беларусь в информационном пространстве» [Электронный ресурс] / В.И. Бричковский // Культура Беларуси: реалии современности : VIII Междунар. науч.-практ. конф., посвящ. Году малой родины в Респ. Беларусь (Минск, 10 октября 2019 г.) : сб. науч. ст. / Бел. гос. ун-т культуры и искусств ; [редкол. : А. А. Корбут и др.]. – Минск, 2019. – С. 78–82. – Режим доступа : <http://repository.buk.by/bitstream/handle/123456789/21174/Integraciya%20kraevdcheskih%20elektronnih%20informacionnih.pdf?sequence=1&isAllowed=y>.

2. Романович, А. Использование баз данных собственной генерации в библиотечно-информационном обслуживании [Текст] / А. Романович // Бібліятэчны свет. – 2019. – № 3. – С. 18–20.

3. Рыкова, В.В. Проблемно-ориентированные базы данных библиотек как инструмент научной коммуникации [Электронный ресурс] / В.В. Рыкова // Научные коммуникации. Научная этика. Инженерная этика : сб. докл. Первой регион.науч. конф. (Омск, 30 сент.–1 окт. 2015 г.) / Омский гос. техн. ун-т, Омск.науч. центр Сиб. отд-ния Рос. акад. наук, Омск. филиал Ин-та математики им. С. Л. Соболева Сиб. отд-ния Рос. акад. наук. – Омск : [ОГТУ], 2015. – С. 101–104. – Режим доступа : [https://elibrary.ru/download/elibrary\\_24806743\\_92836587.pdf](https://elibrary.ru/download/elibrary_24806743_92836587.pdf).

4. Сивенкова, И.Н. Электронные ресурсы библиотеки Могилевского государственного университета продовольствия для образования и научной сферы [Электронный ресурс] / И.Н. Сивенкова // Библиотеки в информационном обществе: сохранение традиций и развитие новых технологий : докл. междунар. науч. конф. (Минск, 3–4 дек. 2014 г.) / Гос. учрежд. «Белорусская сельскохозяйственная библиотека им. И.С. Лупиновича» Нац. акад. наук Беларуси ; [редколлегия: В. В. Юрченко и др.]. – Минск : Ковчег, 2014. – С. 169–175. – Режим доступа : [https://elibrary.ru/download/elibrary\\_25678553\\_91349313.pdf](https://elibrary.ru/download/elibrary_25678553_91349313.pdf).

5. Ходотчук, Н.П. Полнотекстовые базы данных библиотеки Гродненского государственного аграрного университета : создание и анализ их использования [Электронный ресурс] / Н.П. Ходотчук, Н.В. Лакотко, Е.Н. Зюзь // Библиотеки в информационном обществе: сохранение традиций и развитие новых технологий : докл. междунар. науч. конф. (Минск, 3–4 дек. 2014 г.) / Гос. учрежд. «Белорусская сельскохозяйственная библиотека им. И.С. Лупиновича» Нац. акад. наук Беларуси ; [редколлегия: В. В. Юрченко и др.]. – Минска : Ковчег, 2014. – С. 169–175. – Режим доступа : [https://elibrary.ru/download/elibrary\\_25678531\\_77856728.pdf](https://elibrary.ru/download/elibrary_25678531_77856728.pdf).

6. Черкесова, К.И. Использование баз данных электронной периодики в практике работы муниципальных библиотек [Электронный ресурс] / К.И. Черкесова // Наука. Искусство. Культура. – 2016. – Вып. 2 (10). – С. 183–186. – Режим доступа : <https://cyberleninka.ru/article/n/ispolzovanie-baz-dannyh-elektronnoy-periodiki-v-praktike-raboty-munitsipalnyh-bibliotek/viewer>.

#### 4. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

#### **4.1 Методические указания к самостоятельной работе**

Самостоятельная работа студентов позволяет закрепить знания, полученные на занятиях по учебной дисциплине «Базы данных», систематизировать информацию, расширить представления об организации и технологии работ по проектированию, созданию и поддержке функционирования баз данных. Контроль за самостоятельной работой студентов осуществляется в ходе составления и проверки аналитических таблиц и отчетов, разработки электронных презентаций, выполнения и проверки реферативных работ.

Критериями оценки результатов самостоятельной работы студентов являются:

- уровень усвоения студентом учебного материала;
- умение студента использовать теоретические знания при выполнении практических задач;
- уровень сформированности общих и профессиональных компетенций;
- умение пользоваться информационно-коммуникационными технологиями;
- обоснованность и четкость изложения материала;
- уровень оформления работы.

#### **4.2 Тематика самостоятельной работы**

**Тема № 1. Неклассические модели данных (2 часа)**

*Цель:* Закрепить знания об основных понятиях и особенностях представления данных с помощью постреляционной, многомерной, объектно-ориентированной и объектно-реляционной моделей данных.

*Задание:*

1. Проанализировать публикации в профессиональных изданиях и изучить, представленные в них основы и характеристики постреляционной, многомерной, объектно-ориентированной и объектно-реляционной моделей данных.

2. В результате изучения постреляционной, объектно-реляционной, многомерной и объектно-ориентированной моделей данных дать ответы на следующие вопросы:

- Что понимают под постреляционной моделью данных?
- Каковы функциональные особенности постреляционной модели данных?
- Каковы достоинства и недостатки постреляционной модели данных?
- Что понимают под объектно-реляционной моделью? В чем заключается ее отличие от реляционной модели?
- Каковы достоинства и недостатки объектно-реляционной модели данных?
- Что понимают под объектно-ориентированной моделью данных?
- Что относят к основным понятиям объектно-ориентированной модели данных? Как их определяют?
- Какие свойства присущи объектно-ориентированной модели данных?
- Каковы достоинства и недостатки объектно-ориентированной модели данных?
- Что понимают под многомерной моделью данных?
- Какие свойства присущи многомерной модели данных?
- Что относят к основным понятиям многомерной модели данных? Как их определяют?
- Каковы достоинства и недостатки многомерной модели данных?

*Форма контроля:* Реферат в бумажной или электронной форме.

Литература

1. Багуи, Сиха. Объектно-ориентированные базы данных: достижения и проблемы [Электронный ресурс] / Сиха Багуи // Открытые системы. СУБД. – 2004. – № 03. – Режим доступа : <https://www.osp.ru/os/2004/03/184042/>. – Дата доступа : 06.03.2020.

2. Головчинер, М.Н. Особенности объектно-ориентированных СУБД / М.Н. Головчинер // Базы данных: основные понятия, модели данных, процесс проектирования : курс лекций / М.Н. Головчинер. – Томск, 2009. – С. 103–110.

3. Исаченко, А.Н. Модели данных и СУБД : учебное пособие / А.Н. Исаченко, С.П. Бондаренко ; Бел. гос. ун-т. – Минск, 2007. – С. 14–16.

4. Кабанов, В.Ф. Основные принципы обработки данных. Современный аспект : учеб. Пособие / В.Ф. Кабанов, А.В. Бурмистров ; Саратовский нац. ун-т им. Н.Г. Чернышевского. – Саратов, 2018. – Гл. 2. Постреляционные СУБД. – С. 13–23.

5. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика: пер. с англ. / Т. Коннолли, К. Бегг. – 3-е изд. – Москва : Вильямс, 2017. – Гл. 24–27. – С. 927–1104.

6. Советов, Б.Я. Базы данных: теория и практика : учебник для бакалавров / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. – 2-е изд. – Москва : Юрайт, 2014. – С. 168–216.

7. Хомоненко, А.Д. Базы данных: учебник для высших учебных заведений / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев ; под ред. А. Д. Хомоненко. – 6-е изд., доп. – СПб.: КОРОНА-Век, 2009. – С. 34–44.

8. Швецов, В.И. Направления развития баз данных [Электронный ресурс] / В.И. Швецов // Базы данных / В.И. Швецов. – Москва : Национальный открытый университет «ИНТУИТ», 2009. – С. 178–185. – Режим доступа : [http://biblioclub.ru/index.php?page=book\\_view\\_red&book\\_id=234676](http://biblioclub.ru/index.php?page=book_view_red&book_id=234676). – Дата доступа : 14.03.2020.

## **Тема № 2. Нормализация схемы базы данных (6 часов)**

*Цель:* Закрепить знания и сформировать практические навыки нормализации отношений (таблиц) на этапе логического проектирования базы данных.

*Задание:*

1. Выбрать и проанализировать вариант схемы базы данных.

### **Вариант 1**

База данных описывает результаты сессии. Информация должна содержать номер семестра, сведения о студенте (ФИО, группа,

специальность), сведения о сдаваемом предмете(название, семестр), дату сдачи экзамена, оценку и ФИО экзаменатора.

#### Вариант 2

База данных описывает работу библиотеки с читателем. Информация должна содержать сведения о читателе(ФИО, адрес, телефон), информацию о выданной книге (название, автор, издательство) и дату выдачи книги.

#### Вариант 3

База данных описывает работу отдела кадров по ведению личных дел сотрудников. Информация должна содержать сведения о сотрудниках(табельный номер, ФИО, отдел (место работы), должность, оклад), отделах(наименование, ФИО начальника, ФИО сотрудников, количество сотрудников).

#### Вариант 4

База данных описывает работу с заказами некоторой оптовой базы. Информация должна содержать сведения о заказчике (Название фирмы, адрес, телефон), сведения о заказываемом товаре (Наименование, фирма изготовитель, год выпуска, стоимость единицы продукции), а также количество заказанного товара.

#### Вариант 5

База данных описывает формирование фонда сети магазинов некоторой фирмы. Информация должна содержать сведения о магазине(название, адрес, телефон), сведения о поставщике (наименование, адрес, телефон) сведения о товаре (наименование, количество) и дату поставки.

#### Вариант 6

База данных описывает работу с клиентами фирмы по техническому обслуживанию торгового оборудования. Информация должна собираться о мастерах, выполняющих ремонтные работы (ФИО, квалификация, телефон), о магазинах, подающих заявки на ремонт оборудования(наименование оборудования, магазин, адрес, телефон) и о выполнении заказа с указанием даты выполнения и оплате.

#### Вариант 7

База данных описывает работу с вкладчиками в отделении банка. База данных должна содержать сведения о сотрудниках (табельный номер, ФИО, должность), клиентах (ФИО, номер паспорта, наименование вклада, дата вклада, сумма вклада, ФИО сотрудника), вкладах (наименование, минимальный срок, минимальная сумма, процентная ставка).

#### Вариант 8

База данных описывает работу с заказами в авторемонтной мастерской. База данных должна содержать сведения о клиенте(ФИО, адрес), тип работы, оплату и информацию об исполнителе (ФИО, квалификация).

#### Вариант 9

База данных описывает поиск книг в каталоге библиотеки. Информация должна содержать сведения о жанрах(наименование, описание), издательствах (наименование, город), книгах (название, автор, издательство, год издания, жанр).

2. Описать модель предметной области неформальным текстом.

3. Уточнить набор данных.

4. Создать универсальное отношение. Записать схему отношения. Описать содержание атрибутов.

5. Определить первичный ключ отношения.

6. Описать текущее состояние предметной области. Представить в виде таблицы.

7. Проверить соответствие отношения требованиям 1НФ. Если необходимо провести нормализацию отношения. Записать схему отношения и содержание таблицы.

8. Проверить соответствие отношения требованиям 2НФ. Если необходимо провести нормализацию отношения. Записать схему базы данных и содержание таблиц.

9. Проверить соответствие отношения требованиям 3НФ. Если необходимо провести нормализацию отношений. Записать схему базы данных в текстовом виде. Представить схему базы данных и графической виде и содержание таблиц.

*Форма контроля:* Отчет в бумажной или электронной форме. Для построения схемы базы данных в графическом виде можно использовать MS Visio.

### **Тема № 3. Разработка базы данных организации (6 часа)**

*Цель:* Закрепить знания и сформировать практические навыки построения инфологической и даталогической моделей базы данных организации.

*Задание:*

1. Выбрать из таблицы 4 «Варианты заданий для разработки базы данных организации» вариант задания, соответствующий номеру студента в

списке учебной группы.

2. На основании выбранного варианта привести: название организации, цель деятельности организации, структура организации, информационные потребности пользователей (кратко).

3. Привести описание основных сущностей (объектов) предметной области в таблице 4.1. Отбор сущностей производится на основе анализа информационных потребностей. Необходимо привести таблицы описания сущностей (сущностей должно быть не менее 3-х).

Таблица 4.1. Сущности предметной области

№ п/п	Наименование сущности	Краткое описание

4. Привести описание атрибутов (не менее 5-ти) для каждого экземпляра сущности в таблице 4.2. Приводить только те атрибуты сущностей, которые необходимы для формирования ответов на регламентированные и непредусмотренные запросы. Для каждого объекта следует привести таблицы его атрибутов.

Таблица 4.2. Атрибуты сущностей

№ п/п	Наименование атрибута	Краткое описание

5. На основе анализа информационных запросов выявить связи между сущностями. Выявленные связи отразить в таблице 4.3.

Таблица 4.3. Связи предметной области

№ п/п	Наименование связи	Сущности, участвующие в связи	Краткое описание

6. На основании выбранного варианта и таблиц 4.1–4.3:

- описать классы объектов (сущностей) и их свойства;
- расставить существующие связи между сущностями;
- на основании таблицы 4.3. в письменной форме обосновать типы связей (1:1, 1:M и т.д.). При графическом построении инфологической модели следует придерживаться единого масштаба для всей схемы. Все прямоугольники, обозначающие классы объектов, должны быть одного размера. Аналогично, все ромбы с именами связей также должны иметь

одинаковый размер.

7. На основании выбранного варианта и таблиц 4.1–4.3, инфологической модели и нормализации базы данных необходимо:

- провести соответствие ключей для каждой таблицы 4.1–3.3;
- заполнить для каждой таблицы базы данных форму, согласно таблице 4.4.

Таблица 4.4. Структура таблицы для датологической модели

№ п/п	Наименование реквизита	Идентификатор	Тип	Длина	Формат изображения	Ограничения и комментарии

Таблица 4. Варианты заданий для разработки базы данных организации

№ варианта	Условие
Вариант №1	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями. самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить датологическую модель базы данных организации.</li> </ol> <p><i>БД – успеваемость студентов высшего учебного заведения.</i>  БД состоит из следующих таблиц: факультеты, кафедры, учебные группы, студенты, ведомости успеваемости.  <i>Таблица факультеты</i> имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.  <i>Таблица кафедра</i> имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, количество преподавателей.</p>

	<p><i>Таблица учебные группы</i> имеет следующие атрибуты: название группы, год поступления, курс обучения, количество студентов в группе.</p> <p><i>Таблица студенты</i> имеет следующие атрибуты: ФИО студента, группа, год рождения, пол, адрес, город, телефон.</p> <p><i>Таблица ведомости успеваемости</i> имеет следующие атрибуты: группа, студент, предмет, оценка.</p>
Вариант №2	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система супермаркета.</i> БД состоит из следующих таблиц: отделы, сотрудники, товары, продажа товаров, должности.</p> <p><i>Таблица отделы</i> имеет следующие атрибуты: название отдела, количество прилавков, кол-во продавцов, номер зала.</p> <p><i>Таблица сотрудники</i> имеет следующие атрибуты: ФИО сотрудника, отдел, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p><i>Таблица должности</i> имеет следующие атрибуты: название должности, сумма ставки.</p> <p><i>Таблица товары</i> имеет следующие атрибуты: название товара, отдел, страна производитель, условия хранения, сроки хранения.</p> <p><i>Таблица продажа товаров</i> имеет следующие атрибуты: ФИО сотрудника-продавца товара, дата, время, количество, цена, сумма.</p>
Вариант №3	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> </ol>

	<p>2. Описать основные сущности предметной области.</p> <p>3. Расставить существующие связи между сущностями: самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</p> <p>4. Построить инфологическую модель базы данных организации.</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – информационная система военного округа.</i> БД состоит изследующих таблиц: места дислокации, вид войск, части, роты, личныйсостав.</p> <p><i>Таблица вид войск</i> имеет следующие атрибуты: название.</p> <p><i>Таблица места дислокации</i> имеет следующие атрибуты: страна, город,адрес, занимаемая площадь.</p> <p><i>Таблица части</i> имеет следующие атрибуты: номер части, местодислокации, вид войск, количество рот.</p> <p><i>Таблица роты</i> имеет следующие атрибуты: название роты, количество служащих.</p> <p><i>Таблица личный состав</i> имеет следующие атрибуты: ФИО, рота,должность, год рождения, год поступления на службу, выслуга лет,награды, участие в военных мероприятиях.</p>
<p>Вариант №4</p>	<p>На основании выбранного варианта выполнить следующее:</p> <p>1. Выполнить анализ предметной области исследуемой организации.</p> <p>2. Описать основные сущности предметной области.</p> <p>3. Расставить существующие связи между сущностями: самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</p> <p>4. Построить инфологическую модель базы данных организации.</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – информационная система библиотеки.</i> БД состоит из следующихтаблиц: библиотека, фонд библиотеки, тип документов, сотрудники,пополнение фонда.</p> <p><i>Таблица библиотека</i> имеет следующие атрибуты: название, адрес,город.</p>

	<p><i>Таблица фонд библиотеки</i> имеет следующие атрибуты: название фонда, библиотека, количество книг, количество журналов, количество газет, количество альбомов, количество диссертаций, количество авторефератов диссертаций.</p> <p><i>Таблица тип документов</i> имеет следующие атрибуты: название типа.</p> <p><i>Таблица сотрудники</i> имеет следующие атрибуты: ФИО сотрудника, библиотека, должность, год рождения, год поступления на работу, образование, зарплата.</p> <p><i>Таблица пополнение фонда</i> имеет следующие атрибуты: фонд, сотрудник, дата, название источника документоснабжения, тип документов, издательство, дата издания, количество экземпляров.</p>
<p>Вариант №5</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система туристического агентства.</i> БД состоит из следующих таблиц: пансионаты, туры, клиенты, путевки, вид жилья.</p> <p><i>Таблица пансионаты</i> имеет следующие атрибуты: название пансионата, адрес, город, страна, телефон, описание территории, количество комнат, наличие бассейна, наличие медицинских услуг, наличие спа-салона, уровень пансионата, расстояние до моря.</p> <p><i>Таблица вид жилья</i> имеет следующие атрибуты: название (дом, бунгало, квартира, 1-я комната, 2-я комната и т.д.), категория жилья (люкс, полуплюкс, и т.д.), пансионат, описание условий проживания, цена за номер в сутки.</p> <p><i>Таблица туры</i> имеет следующие атрибуты: название тура</p>

	<p>(Европа, Средняя Азия, США и т.д.), вид транспорта, категория жилья на ночь (гостиница, отель, палатка и т.д.), вид питания (одноразовое, двухразовое, трехразовое, завтраки), цена тура в сутки.</p> <p><i>Таблица клиенты</i> имеет следующие атрибуты: ФИО клиента, паспортные данные, дата рождения, адрес, город, телефон.</p> <p><i>Таблица путевки</i> имеет следующие атрибуты: ФИО клиента, название пансионата, вид жилья, дата заезда, дата отъезда, наличие детей, наличие медицинской страховки, количество человек, цена, сумма.</p>
<p>Вариант №6</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система автопредприятия города.</i> БД состоит из следующих таблиц: автотранспорт, водители, маршруты, обслуживающий персонал, гаражное хозяйство.</p> <p><i>Таблица автотранспорт</i> имеет следующие атрибуты: название транспорта (автобусы, такси, маршрутные такси, прочий легковой транспорт, грузовой транспорт и т.д.), количество наработки, пробег, количество ремонтов, характеристика.</p> <p><i>Таблица маршруты</i> имеет следующие атрибуты: название маршрута, транспорт, водитель, график работы.</p> <p><i>Таблица водители</i> имеет следующие атрибуты: ФИО водителя, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p><i>Таблица обслуживающий персонал</i> имеет следующие атрибуты: должность (техники, сварщики, слесари, сборщики и др.), ФИО, год рождения, год поступления на работу, стаж,</p>

	<p>пол, адрес, город, телефон.</p> <p><i>Таблица гаражное хозяйство</i> имеет следующие атрибуты: название гаража, транспорт на ремонте, вид ремонта, дата поступления, дата выдачи после ремонта, результат ремонта, персонал, производящий ремонт.</p>
Вариант №7	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система поликлиники.</i> БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, обслуживающий персонал.</p> <p><i>Таблица отделения</i> имеет следующие атрибуты: название отделения (хирургия, терапия, неврология и т.д.), этаж, номера комнат, ФИО заведующего.</p> <p><i>Таблица врачи</i> имеет следующие атрибуты: ФИО врача, должность, стаж работы, научное звание, адрес, номер отделения.</p> <p><i>Таблица пациенты</i> имеет следующие атрибуты: ФИО пациента, адрес, город, возраст, пол.</p> <p><i>Таблица диагнозы</i> имеет следующие атрибуты: название диагноза, признаки болезни, период лечения, назначения.</p> <p><i>Таблица история болезни</i> имеет следующие атрибуты: ФИО пациента, ФИО врача, диагноз, лечение, дата заболевания, дата выздоровления, вид лечения (амбулаторное, стационарное).</p>
Вариант №8	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями:</li> </ol>

	<p>самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</p> <p>4. Построить инфологическую модель базы данных организации.</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – информационная система больницы.</i> БД состоит из следующихтаблиц: врачи, пациенты, история болезней, операции, лист лечения.</p> <p><i>Таблица врачи</i> имеет следующие атрибуты: ФИО врача, должность, стаж работы, научное звание, адрес.</p> <p><i>Таблица пациенты</i> имеет следующие атрибуты: ФИО пациента,адрес, город, возраст, пол.</p> <p><i>Таблица история болезни</i> имеет следующие атрибуты: ФИО пациента,ФИО врача,диагноз, дата заболевания, дата выздоровления, вид лечения (амбулаторное,стационарное), код операции.</p> <p><i>Таблица лист лечения</i> имеет следующие атрибуты: ФИО пациента, дата лечения,история болезни, лекарства, температура, давление, состояние пациента (тяжелое, среднее, и т.д.).</p> <p><i>Таблица операции</i> имеет следующие атрибуты: описание операции(удаление аппендицита, пластическая операция и т.д.), ФИО врача, датаоперации, ФИО пациента, результат операции.</p>
<p>Вариант №9</p>	<p>На основании выбранного варианта выполнить следующее:</p> <p>1. Выполнить анализ предметной области исследуемой организации.</p> <p>2. Описать основные сущности предметной области.</p> <p>3. Расставить существующие связи между сущностями.</p> <p>4. Построить инфологическую модель базы данныхорганизации.</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – информационная система обслуживания пользователей збиблиотеки.</i> БД состоит изследующих таблиц: библиотека, читальные залы, документы, пользователи,выдача</p>

	<p>документов.</p> <p><i>Таблица библиотека</i> имеет следующие атрибуты: название, адрес, город.</p> <p><i>Таблица читальные залы</i> имеет следующие атрибуты: название читального зала, количество экземпляров документов, количество пользовательских мест, время работы, этаж, количество сотрудников.</p> <p><i>Таблица пользователи</i> имеет следующие атрибуты: ФИО пользователя, категория пользователя, место работы или обучения, возраст, дата регистрации в библиотеке.</p> <p><i>Таблица документы</i> имеет следующие атрибуты: вид документа, название, автор (-ры), издательство, год издания, количество страниц, читальный зал.</p> <p><i>Таблица выдача документов</i> имеет следующие атрибуты: ФИО пользователя, документы, дата выдачи, срок выдачи, наличие залога.</p>
<p>Вариант №10</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система автосалона.</i> БД состоит из следующих таблиц: автомобили, марки автомобилей, сотрудники, продажа автомобилей, покупатели.</p> <p><i>Таблица марки автомобилей</i> имеет следующие атрибуты: название марки, страна производитель, завод производитель, адрес.</p> <p><i>Таблица автомобили</i> имеет следующие атрибуты: название автомобиля, марка автомобиля, год производства, цвет, категория, цена.</p> <p><i>Таблица покупатели</i> имеет следующие атрибуты: ФИО</p>

	<p>покупателя, паспортные данные, адрес, город, возраст, пол.</p> <p><i>Таблица сотрудники</i> имеет следующие атрибуты: ФИО сотрудника, стаж, зарплата.</p> <p><i>Таблица продажа автомобилей</i> имеет следующие атрибуты: дата, ФИО сотрудника, название автомобиля, марка автомобиля, ФИО покупателя.</p>
<p>Вариант №11</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – успеваемость студентов кафедры.</i> БД состоит из следующих таблиц: кафедры, дисциплины, преподаватели, студенты, ведомости успеваемости.</p> <p><i>Таблица кафедры</i> имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, количество преподавателей.</p> <p><i>Таблица преподаватели</i> имеет следующие атрибуты: ФИО преподавателя, название кафедры, год рождения, год поступления на работу, стаж, должность, пол, адрес, телефон.</p> <p><i>Таблица студенты</i> имеет следующие атрибуты: ФИО студента, название кафедры, год рождения, пол, адрес, телефон.</p> <p><i>Таблица дисциплины</i> имеет следующие атрибуты: название дисциплины, название кафедры, количество часов, вид итогового контроля.</p>
<p>Вариант №12</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями:</li> </ol>

	<p>самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</p> <p>4. Построить инфологическую модель базы данных организации.</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – торговая организация.</i> БД состоит из следующих таблиц: торговаяорганизация, торговая точка, продавцы, поставщики, заказыпоставщикам.</p> <p><i>Таблица торговая организация</i> имеет следующие атрибуты: названиеторговой организации, адрес, ФИО директора, налоговый номер.</p> <p><i>Таблица торговая точка</i> имеет следующие атрибуты: название торговойточки, тип (универмаги, магазины, киоски, лотки и т.д.), торговаяорганизация, адрес, ФИО заведующего.</p> <p><i>Таблица продавцы</i> имеет следующие атрибуты: ФИО продавца, торговая точка, должность, год рождения, пол, адрес проживания.</p> <p><i>Таблица поставщики</i> имеет следующие атрибуты: название поставщика, тип деятельности, страна, адрес.</p> <p><i>Таблица заказы поставщикам</i> имеет следующие атрибуты: дата заказа, торговая точка, название поставщика, название товара, количество товара, цена.</p>
<p>Вариант №13</p>	<p>На основании выбранного варианта выполнить следующее:</p> <p>1. Выполнить анализ предметной области исследуемой организации.</p> <p>2. Описать основные сущности предметной области.</p> <p>3. Расставить существующие связи между сущностями: самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</p> <p>4. Построить инфологическую модель базы данных организации.</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – проектная организация.</i> БД состоит из следующих таблиц:отделы, сотрудники, организации, договоры, проектные работы.</p>

	<p><i>Таблица отделы</i> имеет следующие атрибуты: название отдела, этаж, телефон, ФИО начальника отдела.</p> <p><i>Таблица сотрудники</i> имеет следующие атрибуты: ФИО сотрудника, должность(конструкторы, инженеры, техники, лаборанты, прочий обслуживающий персонал), номер отдела, пол, адрес, дата рождения.</p> <p><i>Таблица организации</i> имеет следующие атрибуты: название организации, тип деятельности, страна, адрес, ФИО директора.</p> <p><i>Таблица договоры</i> имеет следующие атрибуты: номер договора, дата заключения договора, название организации, стоимость договора.</p> <p><i>Таблица проектные работы</i> имеет следующие атрибуты: дата начала проектной работы, дата завершения проектной работы, номер договора, отдел.</p>
<p>Вариант № 14</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система военно-морского флота.</i> БД состоит из исследующих таблиц: базы военно-морского флота, части, личный состав, корабли, учения.</p> <p><i>Таблица базы военно-морского флота</i> имеет следующие атрибуты: название базы, географическое расположение, количество частей.</p> <p><i>Таблица части</i> имеет следующие атрибуты: номер части, база флота, место базирования, вид войск (морская авиация, морская пехота и т.д.).</p> <p><i>Таблица личный состав</i> имеет следующие атрибуты: ФИО служащего, часть, должность, год рождения, год поступления на службу, выслуга лет, награды.</p>

	<p><i>Таблица корабли</i> имеет следующие атрибуты: идентификационный номер корабля, название корабля, тип корабля, дата создания, наработка, количество посадочных мест, устройство двигателя (парусное, гребное, паровой, теплоход, турбоход, и т.д. ), тип привода (самоходное, несамоходное), размещение корпуса (подводная лодка, ныряющее, полупогружное, и т.д.)</p> <p><i>Таблица учения</i> имеет следующие атрибуты: номер части, название корабля, дата учения, место проведения, оценка.</p>
<p>Вариант №15</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – туристическая фирма.</i> БД состоит из следующих таблиц: туристы, туристическая группа, состав групп, гостиницы, ведомости продаж.</p> <p><i>Таблица туристы</i> имеет следующие атрибуты: ФИО туриста, паспортные данные, пол, возраст, дети.</p> <p><i>Таблица туры</i> имеет следующие атрибуты: название тура, страна, города, тип передвижения, тип питания, цена тура, тип проживания.</p> <p><i>Таблица туристическая группа</i> имеет следующие атрибуты: название группы, дата отправления, дата прибытия, тур, количество туристов.</p> <p><i>Таблица состав групп</i> имеет следующие атрибуты: дата продажи, ФИО туриста, название группы, цена билета.</p> <p><i>Таблица гостиницы</i> имеет следующие атрибуты: название гостиницы, страна, город, адрес, количество мест, тип гостиницы.</p> <p><i>Таблица ведомость продаж</i> имеет следующие атрибуты: дата,</p>

	название туристической группы, гостиница, общая стоимость.
Вариант №16	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – цирк.</i> БД состоит из следующих таблиц: работники цирка,представления, расписание гастролей, труппа цирка, программа цирка.</p> <p><i>Таблица работники цирка</i> имеет следующие атрибуты: ФИО работника, год рождения, год поступления на работу, стаж, должность(акробат, клоун, гимнаст, музыкант, постановщик, служащий и т.д.), пол,адрес, телефон.</p> <p><i>Таблица представления</i> имеет следующие атрибуты: название представления,режиссер-постановщик, художник-постановщик, дирижер-постановщик,автор, жанр, тип.</p> <p><i>Таблица расписание гастролей</i> имеет следующие атрибуты: название представления, дата начала, дата окончания, места проведения гастролей.</p> <p><i>Таблица труппа цирка</i> имеет следующие атрибуты: название представления, ФИО актера цирка, название роли.</p> <p><i>Таблица программа цирка</i> имеет следующие атрибуты: название представления,дата премьеры, период проведения, дни и время, цена билета.</p>
Вариант №17	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</li> </ol>

	<p>4. Построить инфологическую модель базы данных организации;</p> <p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – аптека.</i> БД состоит из следующих таблиц: лекарства, покупатель, продавец, рецепты, продажа лекарств.</p> <p><i>Таблица лекарства</i> имеет следующие атрибуты: название лекарства, тип (готовое,изготавливаемое), вид (таблетки, капсулы, мази, гели, настойки), цена.</p> <p><i>Таблица покупатель</i> имеет следующие атрибуты: ФИО покупателя, адрес, телефон.</p> <p><i>Таблица продавец</i> имеет следующие атрибуты: ФИО продавца, дата поступления на работу, дата рождения, образование.</p> <p><i>Таблица рецепты</i> имеет следующие атрибуты: номер рецепта, дата выдачи, ФИО покупателя, ФИО врача, диагноз пациента.</p> <p><i>Таблица продажа лекарств</i> имеет следующие атрибуты: дата, название лекарства, количество, номер рецепта, ФИО продавца.</p>
<p>Вариант №18</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельнодобавить в каждую сущность первичные ключи и установить внешниеключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – городская телефонная сеть.</i> БД состоит из следующих таблиц: АТС, абонент, ведомость звонков, прайс АТС, ведомость абонентскойплаты.</p> <p><i>Таблица АТС</i> имеет следующие атрибуты: название АТС, вид (городские, ведомственные, локальные), адрес, количество абонентов.</p> <p><i>Таблица абоненты</i> имеет следующие атрибуты: ФИО</p>

	<p>абонента, вид телефона (основной, параллельный или спаренный), номер телефона, межгород (открыт/закрыт), льгота (да/нет), адрес: индекс, город, район, улица, дом, квартира.</p> <p><i>Таблица ведомость звонков</i> имеет следующие атрибуты: ФИО абонента, дата звонка, время начала звонка, время окончания звонка, межгород (да/нет).</p> <p><i>Таблица прайс АТС</i> имеет следующие атрибуты: название АТС, цена нагородские звонки, цена на междугородние звонки.</p> <p><i>Таблица ведомость абонентской платы</i> имеет следующие атрибуты: ФИО абонента, месяц, год, количество минут на городские звонки, количество минут на междугородние звонки, стоимость, сумма льготы, общая стоимость.</p>
<p>Вариант №19</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – аэропорт.</i> БД состоит из следующих таблиц: работники аэропорта, расписание вылетов, самолеты, бригады самолетов, ведомость продаж билетов.</p> <p><i>Таблица работники аэропорта</i> имеет следующие атрибуты: ФИО работника, год рождения, год поступления на работу, стаж, должность (пилот, диспетчер, техник, кассир, работник службы безопасности, работник справочной службы и др.), пол, адрес, телефон.</p> <p><i>Таблица расписание вылетов</i> имеет следующие атрибуты: название самолета, дата вылета, время вылета, место выбытия, место прибытия, маршрут (начальный и конечный пункты назначения, пункт пересадки), стоимость билета.</p>

	<p><i>Таблица самолеты</i> имеет следующие атрибуты: номер, год выпуска, количество посадочных мест, грузоподъемность.</p> <p><i>Таблица бригады самолетов</i> имеет следующие атрибуты: номер бригады, название самолета, ФИО работника, должность (пилот, техник, обслуживающий персонал).</p> <p><i>Таблица ведомость продажи билетов</i> имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, количество билетов, льготы (пенсионеры, дети-сироты и т.д.), багаж(да/нет), стоимость.</p>
<p>Вариант №20</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – театр.</i> БД состоит из следующих таблиц: работники театра, спектакли, расписание гастролей, труппа спектакля, репертуар театра.</p> <p><i>Таблица работники театра</i> имеет следующие атрибуты: ФИО работника, год рождения, год поступления на работу, стаж, должность (актер, музыкант, постановщик, служащий), пол, адрес, телефон.</p> <p><i>Таблица спектакли</i> имеет следующие атрибуты: название спектакля, режиссер-постановщик, художник-постановщик, дирижер-постановщик, автор, жанр (музыкальная комедия, трагедия, оперетта и др.), тип (детский, молодежный и др.).</p> <p><i>Таблица расписание гастролей</i> имеет следующие атрибуты: название гастроль, дата начала, дата окончания, место проведения, название спектакля.</p> <p><i>Таблица труппа спектакля</i> имеет следующие атрибуты: название спектакля, ФИО актера, название роли.</p> <p><i>Таблица репертуар театра</i> имеет следующие атрибуты:</p>

	название спектакля, дата премьеры, период проведения, дни и время, цена билета.
Вариант №21	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – железнодорожный вокзал.</i> БД состоит из следующих таблиц: работники вокзала, расписание движения поездов, поезда, бригады поездов, ведомость продаж билетов.</p> <p><i>Таблица работники вокзала</i> имеет следующие атрибуты: ФИО работника, год рождения, год поступления на работу, стаж, должность (машинист, диспетчер, проводник, ремонтник подвижного состава, кассир, работник службы подготовки составов и др.), пол, адрес, телефон.</p> <p><i>Таблица расписание движения поездов</i> имеет следующие атрибуты: номер поезда, дата отправления, время отправления, место отправления, дата прибытия, время прибытия, место прибытия, маршрут (начальный и конечный пункты назначения, основные узловые станции), стоимость билета.</p> <p><i>Таблица поезда</i> имеет следующие атрибуты: номер поезда, год выпуска, количество вагонов, тип поезда (общий, скоростной, высокоскоростной).</p> <p><i>Таблица бригады поездов</i> имеет следующие атрибуты: номер бригады, номер поезда, ФИО работник вокзала, должность (машинист, техник, проводник, обслуживающий персонал).</p> <p><i>Таблица ведомость продажи билетов</i> имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер поезда, количество билетов, льготы (пенсионеры, дети-сироты и т.д.), стоимость.</p>

<p>Вариант №22</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> <li>5. Построить даталогическую модель базы данных организации.</li> </ol> <p><i>БД – информационная система высшего учебного заведения.</i></p> <p>БД состоит из следующих таблиц: факультеты, кафедры, преподаватели, дисциплины, учебная нагрузка.</p> <p><i>Таблица факультеты</i> имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.</p> <p><i>Таблица кафедры</i> имеет следующие атрибуты: название кафедры, ФИО заведующего, номер комнаты, номер корпуса, телефон, количество преподавателей.</p> <p><i>Таблица дисциплины</i> имеет следующие атрибуты: название дисциплины, количество часов, цикл дисциплин.</p> <p><i>Таблица преподаватели</i> имеет следующие атрибуты: ФИО преподавателя, название кафедры, год рождения, год поступления на работу, стаж, должность, пол, город.</p> <p><i>Таблица учебная нагрузка</i> имеет следующие атрибуты: ФИО преподавателя, название дисциплины, учебный год, семестр, группы, количество студентов, видитогового контроля.</p>
<p>Вариант №23</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> <li>1. Выполнить анализ предметной области исследуемой организации.</li> <li>2. Описать основные сущности предметной области.</li> <li>3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями.</li> <li>4. Построить инфологическую модель базы данных организации.</li> </ol>

	<p>5. Построить даталогическую модель базы данных организации.</p> <p><i>БД – информационная система супермаркета.</i> БД состоит из следующих таблиц: отделы, клиенты, товары, продажа товаров, поставщики.</p> <p><i>Таблица отделы</i> имеет следующие атрибуты: название отдела, количество прилавков, количество продавцов, номер зала.</p> <p><i>Таблица клиенты</i> имеет следующие атрибуты: название клиента, адрес, вид оплаты.</p> <p><i>Таблица поставщики</i> имеет следующие атрибуты: название поставщика, адрес, страна, вид транспорта, вид оплаты.</p> <p><i>Таблица товары</i> имеет следующие атрибуты: название товара, название отдела, название поставщика, условия хранения, сроки хранения.</p> <p><i>Таблица продажа товаров</i> имеет следующие атрибуты: название клиента, название товара, дата, время, количество, цена, сумма.</p>
--	---

#### **Тема № 4. Решение задач создания запросов инструментарием реляционной алгебры и реляционного исчисления (6 часов)**

*Цель:* Закрепить знания и сформировать практические навыки решения задач создания запросов, используя инструментальные средства и синтаксические правила реляционной алгебры и реляционного исчисления.

*Задание:*

1. Изучить вспомогательный материал по теме в Приложении 1.
2. Описать, в каких обстоятельствах можно воспользоваться каждой из следующих операций реляционной алгебры:
  - Выборка;
  - Создание проекции;
  - Соединение;
  - Присвоение;
  - Вычитание;
  - Пересечение;
  - Деление.
3. Объяснить функции каждого из следующих элементов формулировки

запроса в реляционном исчислении:

- Целевой список;
- Определяющее выражение;
- Булевы операции (И, ИЛИ, НЕ);
- Квантор существования;
- Квантор всеобщности.

4. Пользуясь приведенной реляционной схемой базы данных, определить, какая(-ие) операция(-ии) реляционной алгебры может быть использована для ответа на поставленные вопросы:

КЛИЕНТ (идентификатор\_клиента, имя, доход)

ПЕРЕВОЗКА (номер, идентификатор\_клиента, вес, пункт\_назначения, водитель)

- Какие клиенты имеют годовой доход свыше 5 миллионов рублей?
- Как называется клиент 555?
- Какой пункт назначения перевозки №4566?
- Как называются клиенты, отправлявшие грузы в Иваново?;
- В какие пункты назначения компании с годовым доходом более 10 миллионов рублей отправляли грузы?

5. Дать решение следующих задач создания запросов инструментарием реляционной алгебры:

- Список всех номеров грузов свыше 500 кг.
- Название клиентов с годовым доходом более 10 миллионном рублей.
- Названия городов, получавших грузы свыше 600 кг.
- Имена водителей, доставлявших грузы в Кострому.
- Клиенты, чьи грузы доставлял водитель Петров.
- Клиенты с годовым доходом более 5 миллионов рублей, отправлявших грузы весом менее 1 кг.
- Список клиентов с максимальным годовым доходом.
- Клиенты, чьи грузы перевозились каждым водителем.

6. Сформулировать запросы, указанные в пункте 5, в реляционном исчислении.

7. Над представленными отношениями выполнить теоретико-множественные и/или специальные реляционные операторы. При необходимости обеспечить совместимость отношений по типу.

7.1. Даны два отношения А и В (Таблица А и Таблица В), кортежи которых содержат личные данные абитуриентов, поступающих в университет. Проиллюстрировать выполнение следующих теоретико-

множественных операторов над отношениями А и В: объединение, пересечение и вычитание.

Таблица А

Фамилия	Имя	Отчество	Номер паспорта	Адрес
Иванов	Иван	Иванович		Владимир
Петров	Петр	Александрович		Муром
Сидоров	Михаил	Викторович		Владимир

Таблица В

Фамилия	Имя	Отчество	Номер паспорта	Прописка
Соколов	Иван	Иванович		Москва
Петров	Петр	Александрович		Муром
Сергеев	Алексей	Викторович		Ковров

7.2. Даны два отношения А и В (Таблица А и Таблица В), кортежи которых содержат личные данные абитуриентов, поступающих в университет. Проиллюстрировать выполнение следующих теоретико-множественных операторов над отношениями А и В: объединение, пересечение и вычитание.

Таблица А

Фамилия	Имя	Отчество	Номер паспорта	Адрес
Соколов	Сергей	Викторович		Ковров
Иванов	Иван	Иванович		Владимир
Петров	Петр	Александрович		Муром
Сидоров	Михаил	Викторович		Владимир

Таблица В

Фамилия	Имя	Отчество	Паспортные данные	Прописка
Соколов	Иван	Сергеевич		Москва
Петров	Петр	Александрович		Муром
Соколов	Сергей	Викторович		Ковров
Сергеев	Алексей	Викторович		Ковров

7.3. Дано отношение В (таблица В), кортежи которого содержат информацию о сдаче вступительных экзаменов в университет.

Проиллюстрировать выполнение реляционных операторов, позволяющих ответить на следующие вопросы:

– кто из абитуриентов получил положительные оценки на вступительных экзаменах в университет?;

– когда и по каким предметам проводятся вступительные экзамены в университет?

Таблица В

Номер экзаменационного листа	Предмет	Дата экзамена	Оценка
	Математика	20.06.2006	Отлично
	Информатика	25.06.2006	Хорошо
	Информатика	25.06.2006	Отлично
	Математика	20.06.2006	Хорошо
	Физика	23.06.2006	Удовлетворительно
	Математика	20.06.2006	Хорошо
	Информатика	25.06.2006	Отлично
	Математика	20.06.2006	Хорошо

7.4. Даны два отношения А и В (Таблица А и Таблица В), кортежи которых содержат информацию об абитуриентах и результатах сдачи экзаменов. Проиллюстрировать выполнение реляционных операторов, позволяющих ответить на следующие вопросы:

– какие оценки получил каждый студент на вступительных экзаменах в университет?;

– кто из студентов получил на вступительных экзаменах хотя бы одну оценку «отлично»?;

– кто из студентов получил оценку «отлично» по всем трем вступительным экзаменам?

Таблица А

Номер экзаменационного листа	Фамилия	Имя	Отчество
	Соколов	Сергей	Викторович
	Иванов	Иван	Иванович
	Петров	Петр	Александрович
	Сидоров	Михаил	Викторович

Таблица В

Номер экзаменационного листа	Предмет	Дата экзамена	Оценка
	Математика	20.06.2006	Отлично
	Информатика	25.06.2006	Хорошо
	Информатика	25.06.2006	Отлично
	Математика	20.06.2006	Хорошо
	Физика	23.06.2006	Удовлетворительно
	Математика	20.06.2006	Хорошо
	Информатика	25.06.2006	Отлично
	Математика	20.06.2006	Хорошо

7.5. Даны два отношения А и В (Таблица А и Таблица В), кортежи которых содержат информацию о проходных баллах по специальностям факультета информационных технологий и результатах сдачи экзаменов абитуриентами. Проиллюстрировать выполнение реляционных операторов, позволяющих ответить на следующие вопросы:

– кто из студентов и на какие специальности поступил, исходя из суммы баллов за два экзамена?;

– кто из студентов не поступит на специальность 230101?

Таблица А

Номер абитуриента	Фамилия	Имя	Отчество	Сумма баллов за два экзамена
	Соколов	Сергей	Викторович	
	Иванов	Иван	Иванович	
	Петров	Петр	Александрович	
	Сидоров	Михаил	Викторович	

Таблица В

Специальность	Проходной балл

### 4.3 График контроля за самостоятельной работой студентов

№ п/п	Темы КСР	Формы контроля	Сроки выполнения
1.	Неклассические модели данных	Защита реферата в бумажной или электронной форме	4–5 недели
2.	Нормализация схемы базы данных	Отчет в бумажной или электронной форме. Обсуждение на семинарском занятии	9–10 недели
3.	Разработка базы данных организации	Защита реферата в бумажной или электронной форме	11–12 недели
4.	Решение задач создания запросов инструментарием реляционной алгебры и реляционного исчисления	Отчет в бумажной или электронной форме.	14–15 недели

## 4.4 Вопросы к зачету

### ПЕРЕЧЕНЬ ТЕОРЕТИЧЕСКИХ ВОПРОСОВ

для проведения зачета

1. Понятия «информация», «данные», «информационная система» и их соотношение.
2. Понятие банка данных (БнД). Классификация БнД.
3. Понятие базы данных (БД). Классификация БД.
4. Сетевые БД, архитектура «файл-сервер», «клиент-сервер».
5. Пользователи БД. Требования к БД со стороны внешних пользователей.
6. Понятие системы управления базами данных (СУБД). Требования к СУБД.
7. Компоненты СУБД.
8. Классификация СУБД.
9. Функции СУБД.
10. Языковые средства современных СУБД (DDL, DML, DCL). Общая характеристика.
11. Понятие моделей данных. Общая классификация моделей данных и их характеристика.
12. Иерархическая модель представления данных и ее общая характеристика.
13. Сетевая модель представления данных и ее общая характеристика.
14. Реляционная модель представления данных. Основные понятия и характеристика. Виды связей.
15. Постреляционная модель представления данных и ее общая характеристика.
16. Объектно-ориентированная модель представления данных и ее общая характеристика.
17. Инфологическая модель данных. Элементы. Способ построения.
18. Даталогическая модель данных.
19. Физическая модель данных.
20. Базовые понятия реляционных БД: тип данных, домен, атрибут, кортеж, ключ, отношение, схема отношений.
21. Понятие и цели нормализации БД.

22. Нормальные формы 1НФ, 2НФ, 3НФ, НФБК, нормальные формы более высокого порядка.
23. Жизненный цикл БД. Этапы жизненного цикла БД.
24. Этапы проектирования БД и их процедуры.
25. Общая характеристика СУБД MSAccess. Основные компоненты и возможности системы.
26. Объекты MS СУБД Access.
27. Структура таблиц и типы данных в СУБД MS Access.
28. Алгоритм создания и заполнения таблиц БД в СУБД MS Access.
29. Алгоритм создания схемы данных БД в СУБД MS Access.
30. Алгоритм создания отчетов в СУБД MS Access.
31. Алгоритм создания запросов в СУБД MS Access.
32. Алгоритм создания форм в СУБД MS Access.
33. Алгоритм создания макросов в СУБД MS Access.

#### **4.5 Вопросы к экзамену**

## ПЕРЕЧЕНЬ ТЕОРЕТИЧЕСКИХ ВОПРОСОВ

для проведения экзамена

1. Понятия «информация», «данные», «информационная система» и их соотношение.
2. Понятие банка данных. Классификация банков данных.
3. Понятие базы данных. Классификация баз данных.
4. Сетевые базы данных. Архитектура «файл-сервер» и «клиент-сервер».
5. Пользователи баз данных. Требования к базам данных со стороны внешних пользователей.
6. Понятие и компоненты системы управления базами данных.
7. Функции и классификация систем управления базами данных.
8. Языковые средства современных систем управления базами данных (DDL, DML, DCL). Общая характеристика.
9. Понятие моделей данных. Общая классификация моделей данных и их характеристика.
10. Реляционная модель представления данных. Основные понятия и характеристика. Виды связей.
11. Постреляционная, многомерная, объектно-ориентированная и объектно-реляционные модели представления данных: характеристика, достоинства и недостатки.
12. Проектирование базы данных. Требования к проекту базы данных. Этапы проектирования.
13. Инфологическое проектирование базы данных. Построение ER-диаграммы.
14. Даталогическое проектирование базы данных. Преобразование ER-диаграммы в схему базы данных.
15. Физическое проектирование базы данных, цель и этапы.
16. Базовые понятия реляционных баз данных: тип данных, домен, атрибут, кортеж, ключ, отношение, схема отношений.
17. Нормализации баз данных: понятие и цели. Формы нормализации.
18. Жизненный цикл базы данных. Этапы жизненного цикла базы данных.
19. Общая характеристика системы управления базами данных MS Access. Основные компоненты и возможности системы.
20. Объекты системы управления базами данных MS Access.

21. Структура таблиц и типы данных в системе управления базами данных MS Access.
22. Алгоритм создания и заполнения таблиц базы данных в системе управления базами данных MS Access.
23. Алгоритм создания схемы данных базы данных в системе управления базами данных MS Access.
24. Алгоритм создания отчетов в системе управления базами данных MS Access.
25. Алгоритм создания запросов в системе управления базами данных MS Access.
26. Алгоритм создания форм в системе управления базами данных MS Access.
27. Алгоритм создания макросов в системе управления базами данных MS Access.
28. Язык SQL: понятие, назначение, стандарты и виды.
29. Основные категории команд языка SQL: DDL, DML, DQL, DCL, TPL, CCL.
30. Структура команды языка SQL. Основные предложения языка SQL: FROM, WHERE, INTO, GROUPBY, HAVING, ORDERBY.
31. Типы запросов в языке SQL. Оптимизация SQL-запросов.
32. Сервер базы данных: понятие, структура и требования. Общая характеристика серверов баз данных Oracle 9 (Oracle), MSSQLServer 2000 (MS), Informix (Informix), Sybase (Sybase), Db2 (IBM) и др.
33. Распределенные системы баз данных: понятие, цель, свойства и требования.
34. Безопасность базы данных: понятие, требования. Угрозы безопасности баз данных.
35. Модели обеспечения безопасности баз данных. Типовые и многоуровневая модели.
36. Доступ к базам данных с использованием технологии WWW. Компоненты, способы и средства доступа.
37. Особенности создания и обеспечения доступа к базам данных в библиотеках.
38. Структурно-содержательная и видовая характеристика баз данных крупнейших библиотек Республики Беларусь.
39. Правовая охрана баз данных в зарубежных странах. Содержание авторских прав в отношении баз данных.

40. Правовая охрана баз данных в Республике Беларусь. Содержание авторских прав в отношении баз данных.

РЕПОЗИТОРИЙ БГУКИ

#### 4.6. Тестовые задания

*Вариант 1*

**1. Как называется организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения?**

- а) банк данных
- б) база данных
- в) информационная система
- г) реляционная таблица
- д) СУБД

**2. Какая база данных содержит обширную информацию самого разного типа: текстовую, графическую, звуковую, мультимедийную?**

- а) документальная
- б) сетевая
- в) реляционная
- г) фактографическая
- д) распределенная

**3. Что такое реляционная база данных?**

- а) база данных, разные части которой хранятся на различных ЭВМ компьютерной сети.
- б) базы данных с табличной формой организации
- в) комплекс аппаратно - программных средств для хранения, изменения и поиска информации, для взаимодействия с пользователем
- г) база, которая содержит краткие сведения об описываемых объектах, представленные в строго определённом формате.
- д) организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения.

**4. Как классифицируются базы данных по типу хранимой информации?**

- а) документальные БД
- б) сетевые БД
- в) распределенные БД
- г) иерархические БД
- д) фактографические БД

**5. Указать основные понятия реляционной базы данных?**

- а) таблица
- б) тип данных
- в) поле
- г) домен
- д) первичный ключ

**6. Что такое запись в реляционной БД?**

- а) это информация об одном объекте той реальной системы, которая представлена в таблице реляционной базы данных.
- б) база данных, разные части которой хранятся на различных ЭВМ компьютерной сети
- в) строка прямоугольной таблицы реляционной базы данных
- г) столбец прямоугольной таблицы реляционной базы данных

д) совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения

**7. Какие действия можно выполнить с помощью СУБД?**

- а) создание структуры базы данных
- б) заполнение базы данных информацией
- в) изменение (редактирование) структуры и содержания БД
- г) поиск информации в базе данных и сортировка данных
- д) защита данных и проверка целостности БД

**8. Лицо или группа лиц, отвечающих за выработку требований к БД, ее проектирование, создание, эффективное использование и сопровождение – это**

- а) Администратор базы данных
- б) Диспетчер базы данных
- в) Программист базы данных
- г) Пользователь базы данных
- д) Технический специалист

**9. Модель представления данных – это**

- а) Логическая структура данных, хранимых в базе данных
- б) Физическая структура данных, хранимых в базе данных
- в) Иерархическая структура данных
- г) Сетевая структура данных

**10. Если каждой записи в таблице А могут соответствовать несколько записей в таблице В, а запись в таблице В не может иметь более одной соответствующей ей записи в таблице А, то это связь...**

- а) один-к-одному
- б) многие-к-себе
- в) один-ко-многим
- г) многие-ко-многим

**11. 6. Объект Access, определяемый и используемый для хранения данных:**

- а) таблицы
- б) запросы
- в) формы
- г) отчеты

**12. Различные приложения пользователей, которые формируют запросы к серверу, проверяют допустимость данных и получают ответы – это**

- а) Сервер базы данных

- б) Клиенты
- в) Сеть
- г) Коммуникационное программное обеспечение

**13. По разделению функций между клиентом и сервером выделяют следующие модели архитектуры клиент-сервер:**

- а) модель доступа к удаленным данным
- б) модель сервера баз данных
- в) модель сервера приложений
- г) модель сервера клиентов

**14. Укажите характеристики, относящиеся к СУБД централизованной архитектуры:**

- а) СУБД, база данных и прикладные программы, которые работают с базой данных, функционируют на центральном компьютере
- б) процессы, связанные с обработкой данных, производятся на центральном компьютере
- в) централизованная архитектура СУБД подразумевает доступ из одного узла локальной сети к ресурсам, находящимся на в других узлах
- г) рабочая станция предназначена для работы пользователя и обладает ресурсами, соответствующими потребностям пользователя
- д) сервер локальной сети предоставляет ресурсы рабочим станциям и другим серверам.

**15. Если на компьютере-клиенте располагаются части приложения, реализующие только функции представления, а прикладные функции размещены на стороне сервера, то говорят о модели ...**

- а) доступа к удаленным данным
- б) сервера баз данных
- в) сервера приложений
- г) сервера клиентов

**16. Внутренний уровень архитектуры СУБД**

- а) Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации
- б) Наиболее близок к пользователю, описывает способ размещения данных на устройствах хранения информации
- в) Наиболее близок к пользователю, описывает обобщенное представление данных
- г) Наиболее близок к физическому, описывает способ размещения

данных в логической структуре базы данных

**17. Внешний уровень архитектуры СУБД**

- а) Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации
- б) Наиболее близок к пользователю, предоставляет возможность манипуляции данными в СУБД с помощью языка запросов или языка специального назначения
- в) Для множества пользователей, описывает обобщенное представление данных
- г) Наиболее близок к физическому, описывает способ размещения данных в логической структуре базы данных

**18. Концептуальный уровень архитектуры СУБД**

- а) Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации
- б) Наиболее близок к пользователю, описывает способ размещения данных на устройствах хранения информации
- в) Наиболее близок к пользователю, предоставляет возможность манипуляции с данными
- г) Переходный от внутреннего к внешнему, описывает обобщенное представление данных для множества пользователей

**19. Проектированием БД занимается**

- а) Администратор БД
- б) Программист БД
- в) Пользователь БД
- г) Проектировщик БД

**20. Основная роль администратора баз данных заключается в:**

- а) определении требований к системе
- б) выборе целевой СУБД
- в) планировании разработки БД
- г) разработке приложений

*Вариант 2*

**1. База данных – это...**

- а) совокупность данных, организованных по определенным правилам
- б) совокупность программ для хранения и обработки больших массивов

информации

в) интерфейс, поддерживающий наполнение и манипулирование данными

г) определенная совокупность информации

**2. Набор программных и аппаратных средств, обеспечивающих выполнение действий по определению данных, их обработке и управлению называется...**

а) база данных

б) система управления базой данных

в) компьютерная база данных

г) система управления данными

**3. Какая база содержит краткие сведения об описываемых объектах, представленные в строго определённом формате.**

а) иерархическая

б) сетевая

в) реляционная

г) фактографическая

д) документальная

**4. Как называется база данных, разные части которой хранятся на различных ЭВМ компьютерной сети?**

а) локальная

б) распределенная

в) сетевая

г) иерархическая

д) реляционная

**5. Как классифицируются базы данных по структуре организации данных?**

а) иерархические БД

б) сетевые БД

в) реляционные БД

г) операционные БД

д) справочно-информационные БД

**6. Если в модели каждый сегмент связан с одним или несколькими на более низком уровне, и только с одним на более высоком уровне, то такая модель называется ...**

а) иерархической моделью

б) реляционной моделью

с) сетевой моделью

**7. Наиболее точным аналогом реляционной базы данных может служить:**

- 1) неупорядоченное множество данных
- 2) генеалогическое древо
- 3) двумерная таблица
- 4) трехмерный куб

**8. Каково назначение СУБД MS Access?**

- а) СУБД MS Access позволяет создание базы данных
- б) СУБД MS Access позволяет редактирование БД
- в) СУБД MS Access позволят манипулирование данными
- г) СУБД MS Access ориентирована на пользователя
- д) СУБД MS Access – это система программирования со своим специализированным языком программирования для создания программ обработки данных

**9. В каком режиме создания таблиц в СУБД Access предоставляется набор таблиц, из которых можно создавать таблицы по своему вкусу. Некоторые таблицы могут полностью подойти для вашего приложения. Тип данных и другие свойства полей уже определены.**

- а) режим таблицы
- б) конструктор таблиц
- в) мастер таблиц
- г) импорт таблиц
- д) связь с таблицами

**10. Если запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот, то это связь...**

- а) многие-ко-многим
- б) многие-к-себе
- в) один-ко-многим
- г) один-к-одному

**11. К пользователям БД относятся следующие категории пользователей:**

- а) системный программист, прикладной программист
- б) инженер электронной техники
- в) администратор системы
- г) оператор компьютера, конечный пользователь
- д) прикладные программы, СУБД

**12. Какую структуру имеет схема данных?**

- а) реляционную структуру
- б) сетевую структуру
- в) иерархическую структуру
- г) распределённую структуру
- д) линейную структуру

**13. Сетевая модель представления данных - данные представлены с помощью**

- а) Таблиц
- б) Списков
- в) Упорядоченного графа
- г) Произвольного графа
- д) Файлов

**14. Если на компьютере-клиенте располагаются части приложения, реализующие только функции представления, а прикладные функции размещены на стороне сервера, то говорят о модели ...**

- а) доступа к удаленным данным
- б) сервера баз данных
- с) сервера приложений
- д) сервера клиентов

**15. Укажите принципы системной архитектуры "клиент–сервер":**

- а) система разбивается на две части – клиентскую и серверную
- б) в качестве основного интерфейса между клиентской и серверной частью выступает язык БД SQL
- в) на рабочих станциях–клиентах работает MSAccess
- г) клиентская часть системы при потребности обращается к серверной части
- д) SQL-сервер выполняет обработку данных

**16. Внутренний уровень архитектуры СУБД**

- а) Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации
- б) Наиболее близок к пользователю, описывает способ размещения данных на устройствах хранения информации
- в) Наиболее близок к пользователю, описывает обобщенное представление данных
- г) Наиболее близок к физическому, описывает способ размещения данных в логической структуре базы данных

### **17. Внешний уровень архитектуры СУБД**

- а) Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации
- б) Наиболее близок к пользователю, предоставляет возможность манипуляции данными в СУБД с помощью языка запросов или языка специального назначения
- в) Для множества пользователей, описывает обобщенное представление данных
- г) Наиболее близок к физическому, описывает способ размещения данных в логической структуре базы данных

### **18. Концептуальный уровень архитектуры СУБД**

- а) Наиболее близок к физическому, описывает способ размещения данных на устройствах хранения информации
- б) Наиболее близок к пользователю, описывает способ размещения данных на устройствах хранения информации
- в) Наиболее близок к пользователю, предоставляет возможность манипуляции с данными
- г) Переходный от внутреннего к внешнему, описывает обобщенное представление данных для множества пользователей

### **19. В группу администраторов БД входят:**

- а) Эксперт по языкам запросов
- б) эксперт по прикладным программам
- в) конечные пользователи
- г) администратор данных

### **20. Основная роль администратора баз данных заключается в:**

- а) определении требований к системе
- б) выборе целевой СУБД
- в) планировании разработки БД
- г) разработке приложений

#### **4.7 Перечень рекомендуемых средств диагностики результатов учебной деятельности студентов**

Для выявления уровня учебных достижений студентов рекомендуется использовать следующие средства диагностики:

- выступления на семинарских занятиях;
- устные и письменные опросы в ходе лекционных и практических занятий;
- выполнение лабораторных заданий с использованием компьютера;
- подготовка рефератов и учебных сообщений;
- групповые дискуссии по наиболее сложным вопросам учебной дисциплины;
- выполнение индивидуальных заданий;
- выполнение тестовых заданий;
- подготовка электронных презентаций;
- подготовка электронных отчетов по результатам выполнения практических и лабораторных заданий;
- защита самостоятельно разработанных заданий (проектов, кейсов);
- зачет;
- экзамен.

## 5. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

### 5.1 Учебная программа

#### *Введение*

Предмет учебной дисциплины, его цель, задачи и место в системе профессиональной подготовке специалистов библиотечно-информационной сферы.

Связь учебной дисциплины с другими учебными дисциплинами. Объем, структура, содержание и порядок изучения учебной дисциплины. Формы самостоятельной работы. Система средств диагностики. Учебно-методическое обеспечение учебной дисциплины

Эволюция развития средств компьютерной обработки данных. Современное состояние, тенденции развития теории и практики баз данных.

#### *Тема 1. Основы теории баз данных*

Определение основных понятий: информация, данные, база данных, система управления базами данных, автоматизированная информационная система, предметная область, банк данных, приложения системы управления базами данных, внешние приложения. Децентрализованный и централизованный подход к организации данных. Преимущества и недостатки таких подходов.

Банк данных (БНД) как информационная система. Состав банка данных в узком и широком смысле этого понятия. Состав банка данных как автоматизированной системы. Компоненты банка данных. Отличительные особенности БНД. Классификация БНД.

База данных (БД) как интегрированный ресурс. Объекты и свойства БД. Отличительные особенности БД. Выполняемые операции с БД.

Основные признаки классификации БД. Документальные БД, их виды. Лексикографические и фактографические БД. Интегрированные и распределенные БД: основные характеристики.

Основные категории пользователей БД.

### ***Тема 2. Системы управления базами данных***

Понятие системы управления базами данных (СУБД). Функции СУБД. Требования к СУБД. Программные компоненты (модули) СУБД: процессор запросов, контроллер БД, контроллер файлов, препроцессор языка DML, компилятор языка DDL, контроллер словаря. Типы обращений к СУБД: запросы, модификации, модификации схемы. Признаки классификации СУБД: степень универсальности, функциональность, язык общения и др. Языковые средства СУБД. Язык описания данных (DDL DataDefinitionLanguage). Язык манипулирования данными (DML DataManipulationLanguage). Язык структурированных запросов (SQL StructuredQueryLanguage). Язык запросов по образцу (QBE – QueryBy Example).

Выбор СУБД: основные подходы к выбору СУБД; показатели пригодности; технические характеристики; оценка производительности. Возможности современных СУБД. Интерфейсы, предоставляемые СУБД пользователям.

Особенности функционирования, применения и использования СУБД MicrosoftAccess, MicrosoftFoxProforWindows, MicrosoftVisualFoxPro, BorlanddB IV, Oracle, MySQL, Microsoft SQL Server, Clipper, Informix, Линтер.

Объектно-реляционные СУБД – СУБД третьего поколения. Основные принципы СУБД третьего поколения, обозначенные в «Манифесте систем баз данных третьего поколения».

### ***Тема 3. Архитектура систем управления базами данных***

Трехуровневая система организации БД, предложенная американским комитетом по стандартизации ANSI (AmericanNationalStandardsInstitute). Цель и причины выделения трех уровней. Понятия схемы и подсхемы. Два типа независимости от данных: логическая и физическая.

Внешний уровень представления БД и его содержание. Языки внешнего уровня. Внешние записи и внешняя схема.

Концептуальный уровень представления БД и его содержание.

Внутренний уровень представления БД и его содержание.

Физический уровень представления БД и его содержание.

Архитектура многопользовательских СУБД. Архитектура «клиент-сервер». Понятия «клиент» и «сервер», их функции. Категории приложений, выполняемых над СУБД: приложения, написанные пользователями; приложения, предоставляемые поставщиками (инструментальные средства). Классы инструментальных средств: процессоры языков запросов; генераторы отчетов; электронные таблицы и др. Три функциональных компонента приложения.

Общая модель функционирования архитектуры «клиент-сервер». Двухзвенная и многозвенная архитектуры «клиент-сервер», их суть. Модель «толстого» клиента. Модель «тонкого» клиента. Достоинства и недостатки архитектуры «клиент-сервер».

Модели архитектуры «клиент-сервер» в зависимости от вариантов разделения функций между «клиентом» и «сервером»: модель файлового сервера (FileServer — FS); модель удаленного доступа к данным (RemoteDataAccess — RDA); модель сервера баз данных (DataServer — DBS); модель сервера приложений (ApplicationServer — AS); модель распределенной БД. Достоинства и недостатки моделей архитектуры «клиент-сервер».

#### ***Тема 4. Модели представления данных***

Категории «данные» и «модель данных». Модель данных – ядро БД. Классификация моделей данных. Физические модели. Инфологические модели. Даталогические модели. Тезаурусные, дескрипторные модели и модели, основанные на языках разметки документов, как разновидности документальных моделей.

Классические модели данных: иерархическая, сетевая, реляционная. Формы представления и основные понятия иерархической, сетевой и реляционной моделей.

Постреляционная модель. Особенности представления данных с помощью постреляционной модели. Многомерная и объектно-ориентированная модели. Основные понятия и особенности представления данных с помощью многомерной и объектно-ориентированной моделей. Объектно-реляционная модель как новый вид модели данных.

Достоинства и недостатки иерархической, сетевой, реляционной, постреляционной, многомерной и объектно-ориентированной моделей данных.

Взаимосвязи в модели данных и их типы. Взаимосвязь «один к одному» (между двумя типами объектов; между двумя атрибутами объектов). Взаимосвязь типа «один ко многим» (между двумя типами объектов; между двумя атрибутами объектов). Взаимосвязь типа «многие ко многим» (между двумя типами объектов; между двумя атрибутами объектов). Графическое отражение взаимосвязей различных типов.

### ***Тема 5. Теоретические основы реляционных баз данных***

Общее понятие реляционного подхода к организации БД. Три основных компонента реляционной базы данных по К. Дж. Дейту: структурный, манипуляционный и целостный. Двенадцать требований (правил) Э.Ф. Кодда, которым должна соответствовать реляционная база данных.

Определение основных понятий реляционных БД: тип данных, домен, атрибут, схема отношения, схема базы данных, кортеж, отношение. Фундаментальные свойства отношений: отсутствие кортежей-дубликатов; отсутствие упорядоченности кортежей; отсутствие упорядоченности атрибутов; атомарность значений атрибутов. Виды отношений: именованное отношение, базовое отношение, производное отношение, выражаемое отношение.

Понятие целостности БД. Средства поддержки целостности и случаи ее нарушения. Понятие транзакции.

Таблица как способ представления отношений, ее элементы. Требования к табличной форме представления отношений: конечность; одноярусность заголовка и уникальность имени столбца; несущественность порядка строк, однотипность данных во всех столбцах.

Ключ связи как инструмент объединения данных из разных таблиц. Отношения записей типа «один к одному» (1:1) и типа «один ко многим» (1:M) в двух таблицах реляционной базы данных.

Необходимость нормализации. Понятия нормализации отношений и нормальной формы. Нормальные формы: 1) первая нормальная форма (1NF); 2) вторая нормальная форма (2NF); 3) третья нормальная форма (3NF); 4) нормальная форма Бойса-Кодда (BCNF); 5) четвертая нормальная форма (4NF); 6) пятая нормальная форма (5NF). Приведение базы данных к

нормализованному виду. Достоинства и недостатки нормализованных и ненормализованных реляционных таблиц.

Системный каталог как база метаданных, его назначение и структура. Преимущества реляционных БД.

### ***Тема 6. Базисные средства манипулирования реляционными данными***

Реляционная алгебра: понятие и основная цель. Реляционное исчисление: кортежные переменные и правильно построенные формулы, целевые списки и выражения реляционного исчисления, реляционное исчисление доменов.

Традиционные (бинарные) операции над множествами: объединение; пересечение; разность; декартово произведение. Использование декартового произведения для получения информации из множества взаимосвязанных таблиц.

Специальные (унарные) реляционные операции: выборка; проекция; переименование, деление, соединение. Варианты операций соединения: операция внутреннего соединения; операции левого внешнего соединения и правого внешнего соединения; операция полного внешнего соединения. Свойства соединения.

Понятие реляционного выражения. Использование реляционных выражений. Запросы к базе данных в форме реляционных выражений.

### ***Тема 7. Основы проектирование баз данных***

Основная цель и задачи процесса проектирования БД. Требования к проекту БД. Функции и задачи Администратора БД в процессе проектирования БД. Факторы, влияющие на проектирование базы данных.

Семь этапов жизненного цикла БД: предварительное планирование; проверка осуществимости; определение требований, концептуальное проектирование, логическое проектирование, физическое проектирование, оценка работы и поддержка БД. Главные задачи каждого этапа.

Этапы проектирования БД. Формулировка и анализ требований. Концептуальное проектирование и его процедуры: определение сущностей и их документирование; определение связей между сущностями и их документирование; создание ER-модели предметной области, определение атрибутов и их документирование; определение значений атрибутов и их документирование; определение первичных ключей для сущностей и их документирование; обсуждение концептуальной модели данных с

конечными пользователями. Требования, предъявляемые к концептуальной модели.

Модель «сущность-связь» (ER-модель) как средство моделирования предметной области. Основные понятия ER-диаграммы. ER-диаграмма связи 1:1. ER-диаграмма связи 1:M. ER-диаграмма связи M:N.

Логическое проектирование и его процедуры: выбор модели данных; определение набора таблиц и их документирование; нормализация таблиц; проверка логической модели данных; определение требований поддержки целостности данных и их документирование; создание окончательного варианта логической модели данных и его обсуждение.

Преобразование ER-модели в схему реляционной БД. Правила формирования набора предварительных таблиц из ER-диаграмм. Нормализация таблиц. Приведение таблиц к формам 1НФ, 2НФ, 3НФ. Нормальная форма Бойса-Кодда (НФБК), 4НФ, 5НФ.

Физическое проектирование и его процедуры: проектирование таблиц БД средствами выбранной СУБД; реализация бизнес-правил в среде выбранной СУБД; проектирование физической организации БД; разработка стратегии защиты БД; организации мониторинга функционирования БД и ее настройка.

Взаимосвязь этапов проектирования БД. Критерии оценки проектируемой/спроектированной БД: адекватность, полнота, адаптируемость, универсальность, сложность структуры БД, степень дублирования данных в БД, объем требуемой памяти, скорость обработки информации.

### ***Тема 8. Основы структурированного языка запросов SQL***

Язык SQL: понятие, назначение, стандарты, достоинства и недостатки. Функциональные возможности языка SQL. Место языка SQL в разработке информационных систем, организованных на основе технологии клиент-сервер. Виды языка SQL: интерактивный и вложенный.

Основные категории команд языка SQL: DDL, DML, DQL, DCL, TPL, CCL.

Функции и основные команды DDL: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX. Функции и команды DML: INSERT, UPDATE, DELETE. Функции DQL и его команда SELECT. Общая структура команды SELECT. Функции и команды DCL: GRANT, REVOKE. Функции и команды TPL: BEGIN, COMMIT, ROLLBACK.

Функции и команды CCL: DECLARE CURSOR, OPEN, CLOSE, FETCH INTO, DROP CURSOR.

Структура команды языка SQL: ключевое слово и предложения. Основные предложения языка SQL: FROM, WHERE, INTO, GROUP BY, HAVING, ORDER BY.

Типы данных языка SQL, определенные стандартом: Символьный, Битовый, Точные числа, Округленные числа, Дата/время, Интервал. Специальные символы и знаки пунктуации SQL. Предикаты.

Типы запросов в языке SQL. Создание различных видов запросов в языке SQL. Оптимизация SQL-запросов.

### ***Тема 9. Администрирование и защита баз данных***

Администратор БД (АБД): понятие и функции. Состав группы администраторов БД. Разделение функций администрирования между администратором данных (АД) и АБД. Функциональные обязанности АД и АБД. Задачи администрирования. Разделение задач администрирования между АД и АБД.

Инструментальные средства администрирования БД: язык определения данных; язык манипулирования данными; словарь данных (системный каталог).

Понятие и цель защиты БД. Объекты БД, подлежащие защите: таблицы, представления, хранимые процедуры и триггеры. Права доступа для каждого типа объектов.

Средства защиты БД в различных СУБД: основные и дополнительные. Парольная защита и установление прав доступа к объектам БД как одни из основных средств защиты БД. Уровни прав доступа: просмотр данных, изменение данных, добавление данных, удаление данных. Возможные последствия нарушения защиты БД.

Угрозы безопасности БД: общие и специфические. Требования безопасности БД. Понятие политики безопасности.

Общая классификация уровней безопасности. Четыре класса безопасности в соответствии с «Критериями определения безопасности компьютерных систем»: D, C, B и A.

Принципы обеспечения безопасности БД: проверка полномочий и проверка подлинности (аутентификация). Иерархический характер системы назначения полномочий. Системный администратор или администратор сервера БД как высший уровень системы назначения полномочий.

Программные решения для защиты БД: FortiDB; SafeNet DataSecure; McAfee Database Security; Secret Disk Server NG. Вредоносные программы и их классификация. Загрузочные и файловые вирусы, программы-закладки. Методы обнаружения и удаления вирусов, восстановления программного обеспечения.

Правовое регулирование охраны БД. Определение БД как особого объекта авторского права. Содержание авторских прав в отношении БД. Защита прав создателей БД средствами авторского права.

### ***Тема 10. Современные направления формирования и использования баз данных в библиотеках***

Библиотеки как центры разработки и использования БД. Виды БД, создаваемых и применяемых в библиотеках. Особенности разработки баз данных в библиотеках.

Объектно-ориентированный подход к созданию БД в деятельности библиотек. Основные понятия объектно-ориентированного подхода: объект, атрибут, класс, наследование, инкапсуляция, полиморфизм, метод. Преимущества и недостатки объектно-ориентированного подхода.

Функциональные требования к объектно-ориентированным БД. Языки запросов объектно-ориентированных СУБД. Достоинства и недостатки объектно-ориентированных БД. Преимущества использования объектно-ориентированных БД в деятельности библиотек.

Объектно-реляционный подход к созданию БД в деятельности библиотек. Преимущества и недостатки объектно-реляционного подхода.

Типы данных, поддерживаемые объектно-реляционными БД: простые, сложные, пользовательские. Достоинства и недостатки объектно-реляционных БД. Преимущества использования объектно-реляционных БД в деятельности библиотек.

Архитектура и принципы разработки распределенных БД в библиотеках. Два способа распределения данных: репликация и фрагментация. Преимущества и недостатки репликации и фрагментации. Проблема транзакций данных в распределенных БД.

Структура и требования к распределенной БД. Основные и дополнительные принципы создания и функционирования распределенной БД, сформулированные К. Дж. Дейтом. Организация связей между фрагментами распределенной БД. Модели распределенных БД. Гомогенные и гетерогенные распределенные БД. Выполнение запросов в распределенных

БД.

Архитектура мультимедийных и гипертекстовых БД. Три класса типов данных, поддерживаемых мультимедийными БД: статические, динамические, многомерные. Модель организации данных в гипертекстовых БД.

### ***Тема 11. Перспективные направления формирования и использования баз данных в библиотеках***

Публикация БД в сети Интернет как новая информационная технология в деятельности библиотек. Задачи размещения информации из БД в сети Интернет. Архитектура Web-приложений, публикующих БД. Обзор Web-серверов. Публикация БД с использованием XML: создание и обработка XML-документа, сценарий для отображения XML-документа, формирование XML-документа на основе БД. Размещение данных из XML-документа в БД. Публикация БД средствами MS Access.

Формирование хранилищ БД в библиотеках. Базовая архитектура хранилищ БД. Принципы построения хранилищ БД: предметная ориентация, возможности интеграции, неизменчивость данных, изменения во времени. Технологические решения хранилищ БД: виртуальное хранилище БД; глобальное хранилище БД; корпоративное хранилище БД; встроенное хранилище БД. OLAP-технология – технология комплексного многомерного анализа данных. Применение OLAP-технологии в работе с БД хранилищ.

Темпоральные и дедуктивные БД: принципы функционирования; возможности создания и использования в библиотеках.

Базы знаний как компонент систем искусственного интеллекта. Основные отличия баз знаний от БД. Требования к информации, хранящейся в базе знаний. Структура и технология использования базы знаний. Основные функции базы знаний. Модели функционирования баз знаний: продукционные модели; семантические сети; фреймы; формальные логические модели.

Экспертные системы как наиболее распространенный класс интеллектуальных систем. Структура экспертной системы. Классификация экспертных систем. Коллектив разработчиков экспертных систем: пользователь, эксперт, инженер по знаниям, программист. Возможности использования экспертных систем в библиотеках.

## 5.2 Учебно-методические карты учебной дисциплины для дневной и заочной форм получения высшего образования

### Учебно-методическая карта учебной дисциплины

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов				Количество часов КСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия		
1	2	3	4	5	6	7	8
1	Введение	2					
2	<b>Тема 1.</b> Основы теории баз данных	4			4		проверка результатов выполнения лабораторной работы, выступление на семинарских занятиях, дискуссионное обсуждение в группах

3	<b>Тема 2.</b> Системы управления базами данных	2	2		2	2	выступление на семинарском занятии, проверка результатов выполнения самостоятельной работы,
1	2	3	4	5	6	7	8
							дискуссионное обсуждение в группах
4	<b>Тема 3.</b> Архитектура систем управления базами данных	2				2	дискуссионное обсуждение в группах, проверка результатов выполнения самостоятельной работы
5	<b>Тема 4.</b> Модели представления данных	2	2		4	2	дискуссионное обсуждение в группах, проверка результатов выполнения

							лабораторной работы, проверка результатов выполнения самостоятельной работы, тестирование
--	--	--	--	--	--	--	---

РЕПОЗИТОРИЙ БГУКИ

1	2	3	4	5	6	7	8
6	<b>Тема 5.</b> Теоретические основы реляционных баз данных	4		2	8	2	дискуссия обсуждение в группах, выступление на семинарских занятиях, проверка результатов выполнения лабораторной работы, проверка результатов выполнения самостоятельной работы, тестирование
7	<b>Тема 6.</b> Базисные средства манипулирования реляционными данными	2			6	2	проверка результатов выполнения лабораторной работы, проверка результатов

							выполне ния самостоя тельной работы
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
8	<b>Тема 7.</b> Основы проектирования баз данных	4			6	2	дискусси онное обсужде ние в группах, проверка результат ов выполне ния лаборато рной работы
9	<b>Тема 8.</b> Основы структурированного языка запросов SQL	4	2		4	2	дискусси онное обсужде ние в группах, проверка результат ов выполне ния лаборато рной работы, проверка результат ов выполне ния самостоя тельной работы

10	<b>Тема 9.</b> Администрирование и защита баз данных	2	2		2	2	проверка результатов выполнения практической работы, проверка результатов
1	2	3	4	5	6	7	8
							ов выполнения лабораторной работы, проверка рефератов, тестирование
11	<b>Тема 10.</b> Современные направления формирования и использования баз данных в библиотеках	2		2		2	выступление на семинарском занятии, проверка результатов выполнения лабораторной работы, проверка рефератов
12	<b>Тема 11.</b> Перспективные	2				2	дискуссионное

	направления формирования и использования баз данных в библиотеках						обсуждение в группах, результатов выполнения практической работы, проверка результатов выполнения
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
							самостоятельной работы, тестирование
		<b>Всего</b>	<b>32</b>	<b>8</b>	<b>4</b>	<b>36</b>	<b>20</b>

### 5.3 Основная литература

1. *Войниканис, Е. А.* База данных как объект правового регулирования : учеб. пособие для вузов / Е. А. Войниканис, В. О. Калятин ; Исслед. центр частного права при Президенте Рос. Федерации. – Москва : Статут, 2011. – 174 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=450696>.
2. *Голицына, О. Л.* Базы данных : учебное пособие для студентов учреждений среднего профессионального образования / О. Л. Голицына. – Изд. 2-е изд., испр. и доп. – Москва : ФОРУМ : ИНФРА-М, 2007. – 400 с.
3. *Дьяков, И. А.* Базы данных. Язык SQL : учеб. пособие / И. А. Дьяков ; М-во образования и науки Рос. Федерации, Федер. гос. бюджетноеобразоват. учрежд. высш. профес. образования «Тамбовский государственный технический университет». – Тамбов :Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 82 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=277628>.
4. *Зыков, Р. И.* Системы управления базами данных / Р. И. Зыков. – Москва : Лаборатория книги, 2012. – 162 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=142314>.
5. *Колкова, Н. И.* Общая технология создания библиографических баз данных / Н. И. Колкова // Справочник библиографа / [А. О. Александрова и др.] ; [редкол. : Г. Ф. Гордукалова (науч. ред.), Т. В. Захарчук, Г. В. Михеева (науч. ред.)]. – Изд. 4-е, испр. и доп. – Санкт-Петербург : Профессия, 2014. – Разд.5, § 5.2. – С. 532–549.
6. *Кузнецов, С.* Введение в реляционные базы данных / С. Кузнецов. – 2-е изд., испр. – Москва : Нац. Открытый Ун-т «ИНТУИТ», 2016. – 248 с. – (Основы информационных технологий). – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=429088>.
7. *Медведкова, И.Е.* Базы данных / И.Е. Медведкова, Ю.В. Бугаев, С.В. Чикунов ; М-во образования и науки Рос. Федерации, Федер. гос. бюджетное образоват. учрежд. высш. профес. образования «Воронежский государственный университет инженерных технологий» ; [науч. ред. Г.В. Абрамов]. – Воронеж : [ВГУИТ], 2014. – 105 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=336039>.
8. *Особенности* создания отдельных видов продуктов // Справочник библиографа / [А. О. Александрова и др.] ; [редкол. : Г. Ф. Гордукалова (науч. ред.), Т. В. Захарчук, Г. В. Михеева (науч. ред.)]. – Изд. 4-е, испр. и доп. –

Санкт-Петербург : Профессия, 2014. – Разд.6. – С. 550–607.

9. Распределенные базы данных : учеб. пособие / М-во образования и науки Рос. Федерации, Федер. гос. автономное образоват. учрежд. высш. профес. образования «Северо-Кавказский федеральный университет» ; [авт.-сост. Н. Ю. Братченко]. – Ставрополь : СКФУ, 2015. – 130 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=457594>.

10. Сенченко, П. В. Организация баз данных : учеб. пособие / П. В. Сенченко ; М-во образования и науки Рос. Федерации, Томск. гос. ун-т систем управления и радиоэлектроники, Ф-т дистанц. обучения. – Томск : ТУСУР, 2015. – 170 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=480906>.

11. Сирант, О. В. Работа с базами данных / О. В. Сирант, Т. А. Коваленко. – 2-е изд., испр. – Москва : Нац. Открытый Ун-т «ИНТУИТ», 2016. – 150 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=428978>.

12. Управление данными : учебник / Ю. Ю. Громов, О. Г. Иванова, А. В. Яковлев, В. Г. Однолько ; М-во образования и науки Рос. Федерации, Федер. гос. бюджетноеобразоват. учрежд. высш. профес. образования «Тамбовский государственный технический университет». – Тамбов :Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 192 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=444642>.

13. Электронные коллекции и электронные фонды // Земсков, А. И. Электронная информация и электронные ресурсы : публикации и документы, фонды и библиотеки / А. И. Земсков, Я.Л. Шрайберг ; [под ред. Л. А. Казаченковой]. – Москва : ФАИР, 2007. – Гл. 7. – С. 285–326.

## 5.4 Дополнительная литература

1. Бушило, Л. БД «Ученые Беларуси» как инновационный корпоративный ресурс библиотек Беларуси / Л. Бушило, Т. Кузьминич // Бібліятэчны свет. – 2013. – № 4. – С. 9–11.
2. Когаловский, М. Р. Энциклопедия технологий баз данных : эволюция технологий, технологии и стандарты, инфраструктура, терминология / М. Р. Когаловский – Москва : Финансы и статистика, 2005. – 800 с.
3. Кузьминич, Т. В. Фактографические базы данных Национальной библиотеки Беларуси – источник информации о национальных объектах / Т.В. Кузьминич // Бібліятэчны вестнік : [зб. арт.] / Нац. б-ка Беларусі ; [склад. : Л. Г. Кірухіна, М. Г. Пшыбытка ; рэдкал. : Р. С. Матульскі (старшыня) і інш.]. – Мінск : [НББ], 2010. – Вып. 2. – С. 158–166.
4. Кусютин, Н. И. Характеристика средств администрирования баз данных и перспективы их развития / Н. И. Кусютин. – Москва : Лаборатория книги, 2011. – 99 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=142999>.
5. Лазицкас, Е. А. Базы данных и системы управления базами данных : учеб. пособие / Е. А. Лазицкас, И. Н. Загумённикова, П. Г. Гилевский. – Минск : РИПО, 2016. – 267 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=463305>.
6. Манифест систем объектно-ориентированных баз данных [Электронный ресурс] / М. Аткинсон, Ф. Бансилон, Д. ДеВитт, К. Дитрих, Д. Майер, С. Здоник; пер.: М.Р. Когаловский // Системы управления базами данных. – 1995. – № 4. – С. 142–155. – Режим доступа : [http://citforum.ru/database/classics/oo\\_manifesto/](http://citforum.ru/database/classics/oo_manifesto/).
7. Основы проектирования баз данных в САПР : учеб. пособие / Ю. В. Литовка [и др.] ; М-во образования и науки Рос. Федерации, Федер. гос. бюджетное образоват. учрежд. высш. профес. образования «Тамбовский государственный технический университет». – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 97 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=277807>.
8. Рыкова, В. В. Проблемно-ориентированные базы данных библиотек как инструмент научной коммуникации [Электронный ресурс] / В. В. Рыкова // Научные коммуникации. Научная этика. Инженерная этика : сб. докл. Первой регион. науч. конф. (Россия, Омск, 30 сент.–1 окт. 2015 г.) / Федер. гос. бюджетное образоват. учрежд. высш. профес. образования «Омский

государственный технический университет» и др. – Омск :Изд-во ФГБОУ ВПО «ОГТУ», 2015. – С. 101–104. – Режим доступа :[https://elibrary.ru/download/elibrary\\_24806743\\_30457589.pdf](https://elibrary.ru/download/elibrary_24806743_30457589.pdf).

9. *Системы баз данных третьего поколения* [Электронный ресурс] : манифест / М. Стоунбрейкер, Л. Роув, Б. Линдсей, Д. Грей, М. Кери, М. Броуди, Ф. Бернштейн, Д. Бич ; пер.: М.Р. Колгановский // Системы управления базами данных. – 1995. – № 2. – С. 143–159. – Режим доступа :<http://citforum.ru/database/classics/manifest/>.

10. *СУБД: язык SQL в примерах и задачах* : учеб.пособие / И. Ф. Астахова [и др.]. – Москва :Физматлит, 2009. – 168 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=76768>.

11. *Ходотчук, Н. П.* Полнотекстовые базы данных библиотеки Гродненского государственного аграрного университета: создание и анализ их использования / Н. П. Ходотчук, Н. В. Лакотко, Е. Н. Зюзь // Библиотеки в информационном обществе: сохранение традиций и развитие новых технологий :докл. междунар. науч. конф., Минск, 3-4 декабря 2014 г. / Гос. учрежд. «Белорусская сельскохозяйственная библиотека им. И.С. Лупиновича» Нац. акад. наук Беларуси ; [ред. : В. В. Юрченко и др. ; рец. : Р. Б. Григянец, С. В. Зыгмантович]. – Минск : Ковчег, 2014. – С. 169–175.

12. *Чурбанова, О. В.* Базы данных и знаний. Проектирование баз данных в MicrosoftAccess : учеб.-метод. пособие / О. В. Чурбанова, А. Л. Чурбанов ; М-во образования и науки Рос. Федерации, Северный (Арктический) федер. ун-т им. М. В. Ломоносова. – Архангельск : САФУ, 2015. – 152 с. – То же [Электронный ресурс]. – URL : <http://biblioclub.ru/index.php?page=book&id=436230>.