Д.Е.СКАЧКОВ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МУЗЕЙНОЙ ДЕЯТЕЛЬНОСТИ

АДМИНИСТРАЦИЯ СМОЛЕНСКОЙ ОБЛАСТИ
ДЕПАРТАМЕНТ СМОЛЕНСКОЙ ОБЛАСТИ ПО КУЛЬТУРЕ
ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СМОЛЕНСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ИСКУССТВ»

УТВЕРЖДАЮ:

Проректор по учебной и воспитательной работе _____ Е.В.Горбылева
«_____» _____ 20____г.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 052a556b0039ad5f8c4f749b3ade26ca30
Владелец: Подгузова Елена Евгеньевна
Действителен: с 05.08.2022 по 29.10.2023

Обсуждена на заседании кафедры:
Зав. кафедрой _____ / Е.С. Мертенс
Протокол № 6
От «25» июня 2020г.

Д.Е.СКАЧКОВ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МУЗЕЙНОЙ ДЕЯТЕЛЬНОСТИ

рабочая программа дисциплины для обучающихся по направлению 51.03.04 Музеология и охрана объектов культурного и природного наследия, **направленности –** Культурный туризм и экскурсионная деятельность; **форме обучения:** очной

Год начала подготовки - 2020

Смоленск
2022

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины являются:

Формирование у студентов системных технологических знаний и элементарных практических умений использования в учебной и производственно-практической деятельности актуальных информационных технологий;

Усвоение базовых информационно-технологических терминов и понятий;

Освоение компонентной структуры информационных технологий различного назначения и сферы применения;

Формирование представлений об ассортименте и потребительских свойствах информационных продуктов и услуг;

Осмысление возможных подходов к оценке эффективности информационных технологий и качества информационных продуктов и услуг;

Формирование практических умений идентифицировать информационные технологии, использовать в учебной, научной и практической деятельности информационные продукты различного назначения.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Информационные технологии в музейной деятельности» находится в обязательной части Блока 1 учебного плана.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины.

ОПК – 2 – Способен решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

В результате освоения дисциплины обучающийся должен:

Знать: основные возможности, предоставляемые современными информационно-коммуникационными технологиями для решения стандартных задач профессиональной деятельности с учетом основных требований информационной безопасности; - информационные процессы профессиональной деятельности; основы теории, нормативную базу, составляющие и пути формирования информационной и библиографической культуры.

Уметь: применять информационно-коммуникационные технологии с учетом основных требований информационной безопасности; - осуществлять самодиагностику уровня профессиональной информационной компетентности.

Владеть: навыками применения информационно-коммуникационных технологий с учетом основных требований информационной безопасности; - методами повышения уровня информационной культуры для решения задач профессиональной деятельности.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часов.

Очная форма обучения

№ п/п	Раздел дисциплины	Семестр по очной форме обучения	Недели семестра по очной форме обучения	Виды учебной работы, включая самостоятельную работу и трудоемкость в часах по очной форме обучения				Форма текущего контроля успеваемости (по неделям семестра) форма промежуточной аттестации (по семестрам) по очной форме обучения
				Лекции	Семинары	Практические	Самостоятельная работа	
1	Раздел 1. Прикладные информационные технологии в музейной сфере	2-3	1-18	20		18	36	контрольная работа
2	Раздел 2. Технологии выставочных и экспозиционных проектов. Разработка и создание виртуальных выставок	3	1-18	34		36	72	контрольная работа
			108	**54**		**54**	**108**	

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При организации учебной деятельности используются как традиционные педагогические технологии, так и образовательные технологии, активизирующие деятельностный подход обучающихся к профессиональному образованию.

В соответствии с требованиями ФГОС ВО по направлению подготовки реализация компетентностного подхода предусматривает использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Интерактивные технологии обучения, предполагают организацию обучения как продуктивной творческой деятельности в режиме взаимодействия обучающихся друг с другом и с преподавателем.

5.1. СОДЕРЖАНИЕ РАЗДЕЛОВ (ТЕМ) ДИСЦИПЛИНЫ

Раздел / тема	Семестр по очной форме обучения	Недели семестра по очной форме обучения	Количество часов по очной форме обучения (по заочной форме)				
			Всего	Лекций	Семинаров	Практических	Самостоятельная работа
РАЗДЕЛ 1. ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МУЗЕЙНОЙ СФЕРЕ							
Тема 1. История внедрения новых информационных технологий в музейное дело	2	1-2	**10**	4			6
Тема 2. Современные технические средства в музейном деле	2	3-5	**16**	4		2	10
Тема 3. Обзор и сравнительные характеристики автоматизированных информационных систем в музее	2	6-10	**22**	4		8	10
Тема 4. Основные понятия теории баз данных	2	11-18	**24**	6		8	10
Тема 5. Проектирование БД методом нормализации	3	1-8	**28**	8		10	10
Тема 6. Проектирование БД с использованием семантических моделей	3	8-10	**18**	4		4	10
Тема 7.Язык SQL	3	11-13	**20**	4		6	10
РАЗДЕЛ 2. ТЕХНОЛОГИИ ВЫСТАВОЧНЫХ И ЭКСПОЗИЦИОННЫХ ПРОЕКТОВ. РАЗРАБОТКА И СОЗДАНИЕ ВИРТУАЛЬНЫХ ВЫСТАВОК							
Тема 8. Интернет в музейном деле	3	13-15	**18**	4		4	10
Тема 9. Информационные технологии обработки графической информации	3	15-14	**18**	4		4	10
Тема 10. Технологии создания обработки видеофайлов	3	14-16	**18**	4		4	10
Тема 11. Проекционный мэппинг в деятельности музеев	3	17	**18**	4		4	10
Тема 12. Интернет в музейном деле	3	18	**10**	4		4	2
ИТОГО			**108**	**54**		**54**	**108**

РАЗДЕЛ 1. ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МУЗЕЙНОЙ СФЕРЕ

Тема 1. История внедрения новых информационных технологий в музейное дело

В определении понятия «информационная технология» в качестве ее объекта выступает информация. В современной научной литературе существует множество подходов к определению содержания понятия «информация».

Для наших целей достаточно указать на практическое совпадение содержания таких понятий, как «информация», «сведения» «сообщение», «данные», которые в словарях и энциклопедиях определяются друг через друга. Будем в дальнейшем опираться на достаточно однозначное понимание содержания этих понятий как сведений о чем-либо.

Эти сведения или информация как объект информационных технологий характеризуются формой восприятия или представления и содержательной интерпретацией, а также материальным носителем

Форма восприятия и представления информации определяет основной способ конечного их использования в той или иной сфере деятельности и предполагает один из следующих вариантов:

- текстовая информация;
- аудиоинформация (звуковая);
- видеоинформация (визуальная).

Текстовая информация – это различные виды письменной речи или представления данных с помощью систем специальных знаков (математические и химические формулы, тексты программ и т.п.).

Аудиоинформация – это устная речь, музыка, звуки естественного или искусственного происхождения, системы звуковых сигналов различного назначения.

Видеоинформация – это различного вида образы, воспринимаемые органами зрения (рисунки, схемы, карты, фильмы и т.п.).

Содержательная интерпретация определяет восприятие конкретной информации той или иной формы восприятия и представления в рамках конкретного вида деятельности или решаемой задачи.

Так, текст некоторого документа на английском языке понятен и может быть использован специалистом, знающим английский язык, но не имеет практического смысла для человека, не владеющего указанным языком. Одна и та же математическая формула описывает различные сущности в зависимости от интерпретации операндов, ее составляющих. Одни и те же звуковые сигналы, подаваемые с помощью горна в различных армиях мира, воспринимаются по-разному. Этих примеров достаточно для того, чтобы показать необходимость такой характеристики информации, как ее содержательная интерпретация.

Носитель информации – это материальное воплощение информации той или иной формы восприятия и представления.

В принципе, в качестве носителя информации может выступать любой материальный объект (в том числе и физическое поле той или иной природы), определенные состояния или свойства которого могут рассматриваться как представление информации. Рассмотрим примеры.

Носителями текстовой информации в разное время человеческой истории выступали такие материальные объекты, как поверхность каменных пещер, выделанные шкуры животных, изготовленные из тростника папирусные свитки, берестяная кора, глиняные и деревянные дощечки, ткани и, наконец, наиболее распространенный в этом отношении носитель – бумага. Все эти носители имели то свойство, что в течение определенного времени изменяли свои физические свойства в диапазоне, позволяющем сохранять изображение текста.

Носители аудиоинформации не так разнообразны. Это прежде всего естественная среда, передающая звуковые волны, а также различного рода искусственные материальные объекты, определенные свойства которых позволяют фиксировать, хранить и воспроизводить звуковые колебания (восковые валики, виниловые диски, намагниченные проволока и пленка, магнитные и оптические диски). Естественно, следует упомянуть и электромагнитные поля, позволяющие воспринимать, передавать и воспроизводить звуковые колебания (радио, телефон, телеграф и т.п.).

Носители видеоинформации естественным образом включают в себя все перечисленные выше носители текстовой информации. Кроме того, они включают в себя различного рода фотоматериалы, голографические пластины и прочие материалы, позволяющие фиксировать, хранить и воспроизводить зрительные образы. К носителям видеоинформации следует отнести электромагнитные поля, позволяющие воспринимать, передавать и воспроизводит звуковые колебания (телевидение).

К особым видам носителей информации относят так называемые «электронные». Это не вполне точное название (поскольку в большинстве случаев речь идет о магнитных и оптических носителях) объединяет все виды носителей, которые хранят данные

в виде некоторых объектов (файлов, дисковых томов и т.п.), интерпретация которых с помощью программ, выполняемых компьютером, воспроизводит ту или иную форму информации на соответствующих устройствах.

Целью, или результатом, информационной технологии является целенаправленное изменение свойств информации, определяемое содержанием решаемой задачи или проблемы.

Такие изменения осуществляются с помощью различного рода информационных преобразований.

Каждое такое преобразование характеризуется содержанием, направлением и объемом.

До второй половины 19 века основу информационных технологий составляли перо, чернильница и бухгалтерская книга. Коммуникация (связь) осуществляется путем направления пакетов (депеш). Продуктивность информационной обработки была крайне низкой, каждое письмо копировалось отдельно вручную, помимо счетов, суммируемых так же вручную, не было другой информации для принятия решений.

Начало XVI века – Леонардо да Винчи создал эскиз тридцатиразрядного суммирующего устройства с десятизубными кольцами.

1723г – нем. ученый Кристиан Людвиг Гестен создал арифметическую машину.

1751 г. – Француз Перера изобрел более компактную арифметическую машину.

1820г. – первый промышленный выпуск цифровых счетных машин-арифмометров.

1822г. – англ. Математик Чарльз Бэббидж создал программно-управляемую счетную машину.

На смену «ручной» информационной технологии в конце 19 века пришла «механическая». Изобретение пишущей машинки, телефона, диктофона, модернизация системы общественной почты - все

это послужило базой для принципиальных изменений в технологии обработки информации и, как следствие, в продуктивности работы. По существу «механическая» технология проложила дорогу к формированию организационной структуры существующих учреждений.

Нач. 20 века – появился арифмометр с клавишами для ввода цифр.

40 - 60-е гг 20 века характеризуются появлением «электрической» технологии, основанной на использовании электрических пишущих машинок со съемными элементами, копировальных машин на обычной бумаге,портативных диктофонов. Они улучшили учрежденческую деятельность за счет повышения качества, количества и скорости обработки документов.

1937-1943гг. – вычислительная машина на электро-магнитных реле – «Марк 1».

1947 год – Марк 2.

1943г. – под руководством Джона Мочли и Проспера Эккерта, математика Джона фон Неймана изобретена ламповая вычислительная машина.

1948г. - изобретен транзистор.

1955г. – начали выпускать компьютеры на транзисторах.

1958г. – изобретена первая интегральная микросхема.

1959г. – разработаны решения по созданию микропроцессора.

Появление во второй половине 60-х годов больших производительных ЭВМ на периферии учрежденческой деятельности (в вычислительных центрах) позволило смесить акцент в информационной технологии на обработку не формы, а содержания информации. Это было началом формирования «электронной», или «компьютерной» технологии. Как известно, информационная технология управления должна содержать как минимум 3 важнейших компонента обработки информации: учет, анализ и принятие решений. Эти компоненты реализуются в «вязкой» среде - бумажном «море» документов, которое становится с каждым годом все более необъятным.

1964г. – разработан компьютер 3-го поколения с применением электронных схем.

Сложившиеся в 60-х годах концепции применения автоматизированных систем управления (АСУ) не всегда и не в полной мере отвечают задаче совершенствования управления и оптимальной реализации компонентов информационной технологии (учет, анализ, принятие решений). Методологически эти концепции нередко опираются на представления о неограниченных возможностях «кнопочной» информационной технологии при непрерывном наращивании вычислительной мощности систем АСУ в применении наиболее общих имитационных моделей, которые в ряде случаев далеки от реального механизма оперативного управления.

Название «автоматизированная система управления» не совсем корректно отражает функции, которые такие системы выполняют, точнее было бы «автоматизированные системы обеспечения управления» (АСОУ), ибо в существующих АСУ, понятие «система» не включает решающего звена управления - пользователя. Игнорирование этого принципиального обстоятельства, по видимому, привело к тому, что расширение сети АСУ и повышение мощности их вычислительных средств обеспечили благодаря большим массивам первичных данных улучшение в основном учетных функций управления (справочных, статистических, следящих). Однако учетные функции отражают только прошлое состояние объекта управления и не позволяют оценить перспективы его развития, т.е. обладают низким динамизмом. В других компонентах технологии управления наращивание мощности АСУ не дало ощутимого эффекта. Отсутствие развитых коммуникационных связей рабочих мест пользователя с центральной ЭВМ, ха-рактерный для большинства АСУ пакетный режим обработки данных, низкий уровень аналоговой поддержки - все это фактически не обеспечивает высокого качества анализа пользователями данных статистической отчетности и всего интерактивного уровня аналитической работы. Тем самым эффективность АСУ на нижних ступенях управленческой лестницы, т.е. именно там, где форми-руются информационные потоки, существенно падает вследствие значительной избыточности поступающей информации при отсутствии средств агреги-рования данных. Именно по этой причине, не смотря на ввод дополнительной системы АСУ, с каждым годом возрастает количество работников, занятых учетными функциями: на сегодняшний день шестую часть всех работников аппарата управления составляет учетно-бухгалтерский персонал.

1975г. – на основе процессора Intel 8080 создан первый массовый ПК – Альтаир.

Начиная с 70-х годов сформировалась тенденция перенесения центра тяжести развития АСУ на фундаментальные компоненты информационных технологий (особенно на аналитическую работу) с максимальным применением человеко-машинных процедур. Но по-прежнему вся эта работа проводилась на мощных ЭВМ, размещенных централизованно в вычислительных центрах. При этом в основу построения подобных АСУ положена гипотеза, согласно которой задачи анализа и принятия решений относились к классу формализуемых, поддающихся математическому моделированию. Предполагалось, что такие АСУ должны повысить качество, полноту, подлинность и своевременность информационного обеспечения лиц, принимающих решения, эффективность работы которых будет возрастать благодаря увеличению числа анализируемых задач.

Однако внедрение подобных систем дало весьма отрезвляющие результаты. Оказалось, что применяемые экономико-математические модели имеют ограниченные возможности практического использования: аналитическая работа и процесс принятия решений происходят в отрыве от реальной ситуации и не подкрепляются информационным процессом формирования. Для каждой новой задачи требуется новая модель, а поскольку модель создавалась специалистами по экономико-математическим методам, а не пользователем, то процесс принятия решений происходит как бы не в реальном времени и теряется творческий вклад самого пользователя, особенно при решении нетиповых управленческих задач. При этом вычислительный потенциал управления, сосредоточенный в вычислительных центрах, находится в отрыве от других средств и технологий обработки информации вследствие неэффективной работы нижних ступеней и необходимости непрерывных конверсий информации. Это так же понижает эффективность информационной технологии при решении задач на верхних ступенях управленческой лестницы. К тому же для сложившейся в АСУ организационной структуры технических средств характерны низкий коэффициент их использования, значительные сроки (не всегда выполняемые) проектирования автоматизированных систем и не высокая их рентабельность из-за слабого воздействия результатов автоматизации на эффективность управления.

Август 1984г. – появился IBM PC.

С появлением персональных компьютеров на «гребне микропроцессорной революции» происходит принципиальная модернизация идеи АСУ: от вычислительных центров и централизации управления, к распределенному вычислительному потенциалу, повышению однородности технологии обработки информации и децентрализации управления. Такой подход нашел свое воплощение в системах поддержки принятия решений (СППР) и экспертных системах (ЭС), которые характеризуют новый этап компьютеризации технологии организационного управления по существу - этап персонализации АСУ. Системность - основной признак СППР и признание того, что самая мощная ЭВМ не может заменить человека. В данном случае речь идет о структурной человеко-машинной единице управления, которая оптимизируется в процессах работы: возможности ЭВМ расширяются за счет структуризации пользователем решаемых задач и пополнения ее базы знаний, а возможности пользователя - за счет автоматизации тех задач, которые ранее было нецелесообразно переносить на ЭВМ по экономическим или техническим соображениям. Становится возможным анализировать последствия различных решений и получать ответы на вопросы типа: «что будет, если...?», не тратя времени на трудоемкий процесс программирования.

Важнейший аспект внедрения СППР и ЭС ¾ рационализация повседневной деятельности работников управления. В результате их внедрения на нижних ступенях управления существенно укрепляется весь фундамент управления, уменьшается нагрузка на централизованные вычислительные системы и верхние ступени управления, что позволяет сосредоточить в них вопросы решения крупных долгосрочных стратегических задач. Естественно, что компьютерная технология СППР должна использовать не только персональные компьютеры, но и другие современные средства обработки информации

Концепция СППР требует пересмотра существующих подходов к управлению трудовыми процессами в учреждении. По существу на базе СППР формируется новая человеко-машинная трудовая единица с квалификацией труда, его нормированием и оплатой. Она аккумулирует знания и умения конкретного человека (пользователя СППР) с интегрированными знаниями и умениями, заложенными в ПЭВМ.

1990г. – создается система баз данных Internet.

АИТ в настоящее время можно классифицировать по ряду признаков, в частности:

- способу реализации в автоматизированных информационных системах (АИС);
- степени охвата АИТ задач управления;
- классам реализуемых технологических операций;
- типу пользовательского интерфейса;
- вариантам использования сети ЭВМ;
- обслуживаемой предметной области.

По способу реализации АИТ в автоматизированных информационных системах выделяют традиционно сложившиеся и новые информационные технологии.

Если традиционные АИТ прежде всего существовали в условиях централизованной обработки данных, до массового использования ПЭВМ были ориентированы главным образом на снижение трудоемкости при формировании регулярной отчетности, то новые информационные технологии связаны с информационным обеспечением процесса управления в режиме реального времени.

Новая информационная технология – это технология, которая основывается на применении компьютеров, активном участии пользователей (непрофессионалов в области программирования) в информационном процессе, высоком уровне дружественного пользовательского интерфейса, широком использовании пакетов прикладных программ общего и проблемного назначения, доступе пользователя к удаленным базам данных и программам благодаря вычислительным сетям ЭВМ.

По степени охвата АИТ задач управления выделяют электронную обработку данных, когда с использованием ЭВМ без пересмотра методологии и организации процессов управления ведется обработка данных с решением отдельных экономических задач, иавтоматизацию управленческой деятельности.

Во втором случае вычислительные средства, включая суперЭВМ и ПЭВМ, используются для комплексного решения функциональных задач, формирования регулярной отчетности и работы в информационно-справочном режиме для подготовки управленческих решений.

К этой же группе могут быть отнесены АИТ поддержки принятия решений, которые предусматривают широкое использование экономико-математических методов, моделей и ППП для аналитической работы и формирования прогнозов, составления бизнес-планов, обоснованных оценок и выводов по изучаемым процессам, явлениям производственно-хозяйственной практики.

К названной группе относятся и широко внедряемые в настоящее время АИТ, получившие название электронного офиса и экспертной поддержки решений. Эти два варианта АИТ ориентированы на использование последних достижений в области интеграции новейших подходов к автоматизации работы специалистов и руководителей, создание для них наиболее благоприятных условий выполнения профессиональных функций, качественного и своевременного информационного обслуживания за счет полного автоматизированного набора управленческих процедур, реализуемых в условиях конкретного рабочего места и офиса в целом.

Электронный офис предусматривает наличие интегрированных пакетов прикладных программ, включающих специализированные программы и информационные технологии, которые обеспечивают комплексную реализацию задач предметной области. В настоящее время все большее распространение приобретают электронные офисы, оборудование и сотрудники которых могут находиться в разных помещениях. Необходимость работы с документами, материалами, базами данных конкретной организации или учреждения в домашних условиях, в гостинице, транспортных средствах привела к появлению АИТ виртуальных офисов. Такие АИТ основываются на работе локальной сети, соединенной с территориальной или глобальной сетью. Благодаря этому абонентские системы сотрудников учреждения независимо от того, где они находятся, оказываются включенными в общую для них сеть.

Автоматизированные информационные технологии экспертной поддержки составляют основу автоматизация труда специалистов-аналитиков. Эти работники кроме аналитических методов и моделей для исследования складывающихся в рыночных условиях ситуаций по сбыту продукции, услуг, финансового положения предприятия, фирмы, финансово-кредитной организации вынуждены использовать накопленный и сохраняемый в системе опыт оценки ситуаций, т.е. сведения, составляющие базу знаний в конкретной предметной области. Обработанные по определенным правилам такие сведения позволяют подготавливать обоснованные решения для поведения на финансовых и товарных рынках, вырабатывать стратегию в областях менеджмента и маркетинга.

По классам реализуемых технологических операций АИТ рассматриваются по существу в программном аспекте и включают:

- текстовую обработку;
- электронные таблицы;
- автоматизированные банки данных;
- обработку графической и звуковой информации;
- мультимедийные и другие системы.

Особое внимание следует обратить на прикладную функциональность, которая, собственно и характеризует применение информационных технологий в музейной деятельности . Музейная тематика - специфическая и довольно узкая предметная область. Соответственно музейные системы не получили такого широкого распространения как, например, бухгалтерские, кадровые, инженерно-расчетные, статистические и т.п. Сейчас мы не будем на них останавливаться и рассмотрим использование информационных технологий, непосредственно связанных с выполнением музейных функций. Хотя, конечно, надо отметить, что в будущем в музеях будут функционировать интегрированные информационные системы, в которых все задачи будут взаимоувязаны. Почему сейчас даже не идет речь об интегрированных информационных системах, учитывающих все аспекты деятельности организации? Ответ неоднозначный. В каких-то вопросах технологии еще не «доросли» до интегрированных решений, а где-то пользователи не в состоянии по самым разным причинам применять новые достижения современной техники и технологий. Интегрированные решения для крупных предприятий на Западе существуют более 10 лет, для малого и среднего бизнеса интегрированные решения и там только начинают разрабатываться, а уж для учреждений культуры и у них, и у нас - это дело будущего.

Персональные компьютеры достаточно широко стали использоваться у нас в стране с начала 1990-х годов. Их применение в сфере культуры было связано с попытками создания автоматизированных информационно-справочных систем, которые могли бы использоваться всеми сотрудниками музея в

повседневной практической работе, экспозиционной, выставочной, экскурсионной и пр. для ускоренного поиска самой различной информации о музейных предметах.

В середине 1970-х годов ведущие музеи страны - Государственный Эрмитаж, Третьяковская галерея и т.д. - делают первые попытки осмысления и практического использования новых возможностей компьютеров.

С середины 1990-х годов компьютеры стали вполне доступны. К этому времени столичные музеи накопили опыт использования компьютеров, и эта информация стала поступать в регионы. В то же время существенно изменились и сами возможности, предоставляемые средствами вычислительной техники.

Можно сказать, что прошедшие 10-15 лет были «периодом первоначального накопления капитала», периодом освоения новых технологий, внедрения их в рабочую практику музеев. И если говорить о серьезных специализированных информационных музейных системах, то начиналось все, несомненно, с автоматизированных систем для целей учета, хранения и изучения коллекций.

Целью создания любого музея является хранение, изучение и популяризация коллекции.

У всех музеев, обратившихся к новым информационным технологиям, есть проблемы самого разного плана.

Переход на автоматизированную работу никогда не означал простого переноса ручного труда на компьютер. При внедрении автоматизированных систем учета, хранения и изучения коллекций, как правило, возникают трудности со структурированием информации, со стандартизацией терминологии и с выработкой классификационных признаков. Нормативная база для описания, атрибуции музейных предметов недостаточно проработана, а для музеев технического профиля отсутствует.

Музейные информационные системы нуждаются в технической поддержке, а музейные работники - в консультациях со стороны специалистов по информационным технологиям. Так не бывает, чтобы разработчик музейной информационной системы установил ее, обучил работников музея и ушел. Всегда есть потребность в постоянном курировании пользователей, особенно «новоиспеченной» информационной системы, со стороны специалистов по информационным технологиям, что приводит к необходимости заключения договоров на техническую поддержку и сопровождение информационной системы. Это не всегда музею «по плечу», так как упомянутые затраты относятся к текущим расходам. Ведь сами информационные системы, как правило, приобретаются музеями в рамках грантов или на спонсорские средства. 1/1 вообще, если говорить о финансовом бремени обслуживания информационных систем, то оно для музеев непосильно тяжелое. Специалисты по информационным технологиям - одна из самых востребованных и высокооплачиваемых категорий работников на рынке труда.

Постоянно увеличивающиеся финансовые затраты ждут всех вступивших на путь компьютеризации и информатизации. Это связано не только с текущими затратами на обслуживание. Перспективное развитие, наращивание функций никогда не поспевает за потребностями обновления быстро устаревающего аппаратного и программного обеспечения. Очевидно, что те, кто идут первыми, очень быстро оказываются «последними», -аппаратное и программное обеспечение устаревает. Начинающие работать на самых новых средствах сразу автоматически опережают вчерашних «первооткрывателей». На сегодняшний день можно констатировать, что наш возрожденный музей оказался в определенном смысле в «передовиках», но это не дает нам оснований для спокойного существования. Мы понимаем, что нужно вкладывать массу средств и усилий, чтобы постоянно идти в ногу с прогрессом в этой области. К этому обязывает и тот факт, что наш музей связи тематически связан с 1/1КТ.

Человеческий фактор как проблема всегда присутствует при внедрении автоматизированных систем, информационных технологий. На каждом этапе внедрения информационных технологий в деятельность музеев ключевую роль играют разные люди или группы людей.

На первом, стартовом этапе очень важна психологическая готовность со стороны руководства музея. Это, пожалуй, один из самых сложных этапов внедрения системы. Во-первых, несмотря на относительное удешевление компьютерной техники, информационных технологий, затраты на них остаются еще достаточно значительными, и от руководства требуется известная смелость вкладывать деньги в малопродуктивное, с точки зрения сиюминутных задач, дело. Во-вторых, очень важно организовать работу всех в единой команде с целью введения единой терминологии, структурирования информации, выработки классификационных принципов и т.д. При этом важную роль играет широкое обсуждение в музейном коллективе терминологии, структуры описаний и принятие компромиссных решений, своего рода «правил игры», без которых зайдешь в тупик. То есть речь идет о выработке коллективных решений, и без лидера, идейного руководителя, здесь не обойтись.

Первыми пользователями автоматизированной системы учета и хранения становятся сотрудники этой службы. Большую роль для успешности внедрения играет выбранная стратегия заполнения базы. Существуют две крайние точки зрения: согласно первой сначала вся информация должна быть выверена и только потом занесена в базу данных. Другая, крайняя точка зрения состоит в том, чтобы вносить в базу те сведения, которые зафиксированы о коллекции на момент начала формирования базы, без необходимости

проверки этих сведений. Стремиться нужно, как всегда, к золотой середине, что не всегда получается, а споры и разногласия на эту тему подчас неразрешимы.

Создание единого музейного информационного ресурса требует изменения индивидуализированной психологии, присущей многим музейным работникам. Сложившаяся в музеях практика работы такова, что люди не стремятся делиться своими наработками, открытиями с широким кругом коллег под предлогом ложно понимаемого авторского права. Работа в автоматизированной системе основана на психологии общей работы. Информация в базу поступает различными путями при проведении различных музейных операций. Таким образом, порой невозможно разделить, кто какую информацию ввел. Вместо разрозненной «своей» информации появляется общий информационный ресурс, качество которого зависит от усилий всех. Создание общего ресурса требует увеличения ответственности каждого за вводимую информацию, нов то же время становится мощным стимулирующим фактором повышения эффективности научной работы и дает возможность большему числу исследователей заниматься обработкой коллекции, одновременно пополняя общий ресурс.

Необходимо отметить еще один момент, связанный с человеческим фактором. До более или менее полного наполнения базы (всей коллекции музея или хотя бы значимой ее части) ее невозможно использовать эффективно. Таким образом, в момент заполнения базы на музейных сотрудников ложится дополнительная нагрузка, которая весьма существенна, а преимущества новой системы начинают ощущаться не сразу. Это работа на перспективу, которая может растянуться на годы.

Одним из обязательных условий работы с новыми информационными технологиями является постоянное получение и осмысление музейными работниками новой информации о технологиях с позиции пользователей, обмен опытом внедрения с коллегами из других музеев.

Профессиональные сообщества и форумы музейных работников по информационным технологиям

Наиболее известные профессиональные сообщества и форумы музейных работников по информационным технологиям: АДИТ, CIDOC, EVA.

В начале 1980-х годов Международный совет музеев ICOM заявил о необходимости внедрения информационных технологий в деятельность музеев. В совете музеев было создано специализированное подразделение - комитет по документации (International Committee for Documentation - CIDOC), куда вошли представители международных организаций и отдельные ученые, профессионально занимающиеся проблемами музейной информатики. CIDOC объединяет около 800 специалистов из 65 стран мира. Ежегодно на международных конференциях CIDOC обсуждаются общие тенденции развития информационных технологий и обмениваются опытом работы. Особое внимание уделяется национальным программам в области информатизации музейной деятельности. В последние годы появились международные проекты, объединяющие национальные программы различных стран в согласованное движение по развитию и использованию музейных информационных ресурсов.

В результате сотрудничества с этой организацией наших соотечественников в мае 1996 года в России была учреждена Ассоциация по документации и новым информационным технологиям в музеях (АДИТ). Инициаторами выступили три крупнейших московских музея - Московский Кремль, Пушкинская и Третьяковская галереи.

Международные конференции EVA (Electronic Imaging & the Visual Arts) - «Электронные изображения и визуальные искусства» - организуются фирмой VASARI Enterprises (Великобритания) с 1989 года по поручению и при финансовой поддержке Комиссии Европейского сообщества. Каждый год проводится «большая EVA» в Великобритании, а также несколько конференций в других европейских странах, в том числе ежегодно в России. В последние годы темы конференций становятся значительно шире, чем название «Электронные изображения и визуальные искусства».

Главное в деятельности таких ассоциаций и форумов - непосредственное общение, обмен практическими соображениями по поводу использования новых технологий. Это чрезвычайно важно для осмысления опыта, не только своего, но и чужого. Результат-объединение усилий, с мастерством и любовью создаваемые новые экспозиции, новые музейные программы, новые мультимедийные продукты.

Тема 2. Современные технические средства в музейном деле

Современные технические средства в музейном деле: аудиофоновизуальные средства; цифровые фотокамеры; сканеры; мультимедийные проекторы; мультимедийные экраны и т.п. Основные технические характеристики.

Термин «мультимедиа» происходит от соединения слов «малти» (много) и «медиа» (среда). Дословно «мультимедиа» означает «многие среды», однако более корректно определять феномен мультимедиа как «полисреда», т.к. это единое пространство, в нерасчлененном виде, представляющее различные виды и формы информации.

Мультимедиа - это особый вид компьютерной технологии, которая объединяет в себе как традиционную, статическую, визуальную информацию (текст, графику), так и динамическую (речь, музыку, видеофрагменты, анимацию). Таким образом, в понятие «мультимедиа» включается широкий спектр

информационных технологий, использующих различные программные и технические средства с целью наиболее эффективного воздействия на человека (ставшего одновременно и читателем, и слушателем, и зрителем).

Благодаря применению в мультимедийных услугах одновременного воздействия графической, аудиальной (звуковой) и визуальной информации, эти средства обладают большим эмоциональным зарядом и активно включаются как в индустрию развлечений, так и в практику информационных учреждений и в домашний досуг. В силу многозначности термина под мультимедиа понимается и мультимедийная программа, и продукт, сделанный на основе мультимедийной технологии, и компьютерное оснащение.

Использование технологии мультимедиа существенно расширило круг задач, решаемых в музеях с помощью компьютера и информационных технологий. Современный музей немыслим без мультимедийного оборудования.

Рассмотрим использование мультимедийного оборудования на примере Центрального музея связи имени А.С. Попова. Концепция музея - показать не только историю техники, но и современные достижения, в том числе в области информационных технологий и мультимедийного оборудования. С учетом этого и строилась экспозиция музея[4].

Мультимедийное оборудование дает возможность оживить подлинные старинные аппараты, продемонстрировать архивные киноматериалы и видеоматериалы, предоставить более подробную, расширенную информацию об экспонатах, о музее для самостоятельного изучения, рассказать о возможностях новых технологий. Все это оборудование (проекторы, сенсорные экраны, плазменные панели, WEB-камеры, управляемые с компьютеров) в музее условно называется экспозиционными автоматизированными рабочими местами. Вообще, их можно было бы назвать и более привычным для музеев термином «информационный киоск», если подразумевать их назначение, но мультимедийные средства разнообразны и конструктивно отличаются друг от друга. Они включены в сеть связи музея, и есть возможность управлять ими централизованно. Поэтому остановились на термине автоматизированное рабочее место посетителя».

Этих мест на сегодняшний день - около 50. Наполнить интересной тематической информацией такое количество мультимедийных средств в очень короткие сроки было просто невозможно. В музее реализован нетрадиционный для музеев вариант: создана гибкая сетевая информационная система со специализированным программным обеспечением и рядом дизайнерских шаблонов для исторической части экспозиции и для экспозиции «Современная связь». Эта система позволяет быстро обновлять все, что демонстрируют мультимедийные средства: текст, «картинки», презентации, видеоматериалы. Имеется удобный административный интерфейс, позволяющий оперативно управлять контентом (содержанием). Для работы с этим интерфейсом не требуются навыки программирования. Существует несколько специально разработанных шаблонов и методика работы с ними. Механизм работы с административным интерфейсом, аналогичный описанному выше, используется для обновления текстовой и мультимедийной информации на сайте музея.

Во входной зоне нашего музея установлены информационные киоски, играющие роль электронного путеводителя.

У музея есть технические возможности организовывать выставки мультимедийных продуктов. Учитывая специфику музея, акцент делается на технологиях, которые были использованы при создании того или иного мультимедийного продукта. В качестве примера можно привести выставку «Информационные технологии в сфере культуры», приуроченную к октябрьской выставке «ИНФОКОМ-2004». Выставка была организована совместно с Государственным Эрмитажем, Государственным Русским музеем и ОАО «Альт-Софт». Мультимедийные продукты этих организаций показывают, что информационные технологии способствуют популяризации и сохранению культурного наследия.

Мультимедийные средства, их возможности (например, проекционное изображение на стену сюжетов по тематике экспозиции) и конструктивные элементы используются в музее связи для дизайнерского оформления экспозиции.

Тема 4. Основные понятия теории баз данных

Базу данных (БД) можно определить как унифицированную совокупность данных, совместно используемую различными задачами в рамках некоторой единой автоматизированной информационной системы (ИС).

Теория управления базами данных как самостоятельная дисциплина начала развиваться приблизительно с начала 50-х годов двадцатого столетия. За это время в ней сложилась определенная система фундаментальных понятий. Приведем некоторые из них.

Предметной областью принято называть часть реального мира, подлежащую изучению с целью организации управления в этой сфере и последующей автоматизации процесса управления. В рамках данной книги для нас в первую очередь представляют интерес предметные области, так или иначе связанные со сферой экономики и финансов.

Объектом называется элемент информационной системы, сведения о котором хранятся в базе данных. Иногда объект также называют сущностью (от англ, entity). Классом объектов называют их совокупность, обладающую одинаковым набором свойств.

Атрибут – это информационное отображение свойств объекта. Каждый объект характеризуется некоторым набором атрибутов.

Ключевым элементом данных называются такой атрибут (или группа атрибутов), который позволяет определить Значения других элементов-данных. Запись данных (англ, эквивалент record) - это совокупность значений связанных элементов данных.

Первичный ключ – это атрибут (или группа атрибутов), который уникальным образом идентифицируют каждый экземпляр объекта (запись). Вторичным ключом называется атрибут (или группа атрибутов), значение которого может повторяться для нескольких записей (экземпляров объекта). Прежде всего вторичные ключи используются в операциях поиска записей.

Процедуры хранения данных в базе должны подчиняться некоторым общим принципам, среди которых в первую очередь следует выделить:

- целостность и непротиворечивость данных, под которыми понимается как физическая сохранность данных, так и предотвращение неверного использования данных, поддержка допустимых сочетаний их значений, защита от структурных искажений и несанкционированного доступа;
- минимальная избыточность данных обозначает, что любой элемент данных должен храниться в базе в единственном виде, что позволяет избежать необходимости дублирования операций, производимых с ним.

Программное обеспечение, осуществляющее операции над базами данных, получило название СУБД – система управления базами данных. Очевидно, что его работа должна быть организована таким образом, чтобы выполнялись перечисленные принципы.

Модели организации данных

Набор принципов, определяющих организацию логической структуры хранения данных в базе, получил название модели данных. Модели баз данных определяются тремя компонентами:

- допустимой организацией данных;
- ограничениями целостности;
- множеством допустимых операций.

В теории систем управления базами данных выделяют модели четырех основных типов: иерархическую, сетевую, реляционную и объектно-реляционную.

Терминологической основой дляиерархической и сетевой моделей являются понятия: атрибут, агрегат и запись. Под атрибутом (элементом данных) понимается наименьшая поименованная структурная единица данных. Поименованное множество атрибутов может образовывать агрегат данных. В некоторых случаях отдельно взятый агрегат может состоять из множества экземпляров однотипных данных, или, как еще говорят, являться множественным элементом. Наконец, записью называют составной агрегат, который не входит в состав других агрегатов. В иерархической модели все записи, агрегаты и атрибуты базы данных образуют иерархически организованный набор, то есть такую структуру, в которой все элементы связаны отношениями подчиненности, и при этом любой элемент может подчиняться только одному какому-нибудь другому элементу. Такую форму зависимости удобно изображать с помощью древовидного графа (схемы, состоящей из точек и стрелок, которая связна и не имеет циклов). Пример иерархической структуры базы данных приведен на рис. 1.

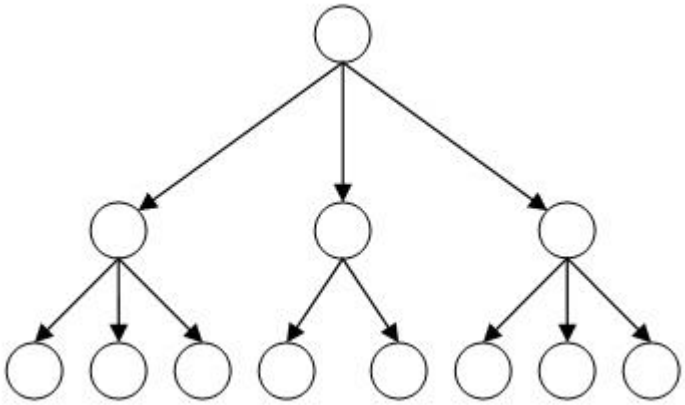

Рис. 1. Схема иерархической модели данных

Типичным представителем семейства баз данных, основанных на иерархической модели, является InformationManagementSystem (IMS) фирмы IBM, первая версия которой появилась в 1968 г. Концепция сетевой модели данных связана с именем Ч. Бахмана. Сетевой подход к организации данных

является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков (рис. 7.2).

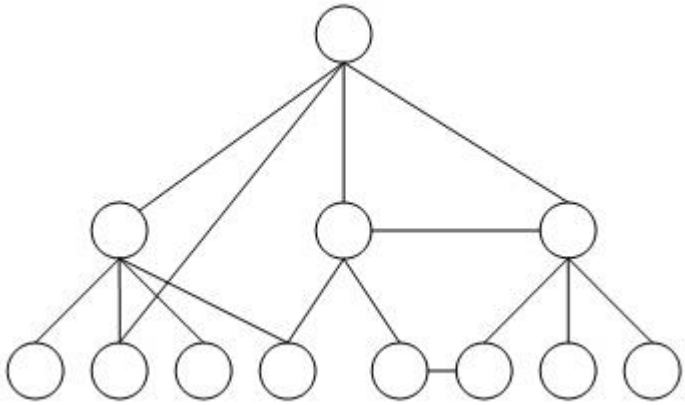

Рис. 2. Схема сетевой модели данных

Сетевая БД состоит из набора записей и набора связей между этими записями, точнее, из набора экземпляров записей заданных типов (из допустимого набора типов) и набора экземпляров из заданного набора типов связи. Примером системы управления данными с сетевой организацией является IntegratedDatabaseManagementSystem (IDMS) компании CullinetSoftwareInc., разработанная в середине 70-х годов. Она предназначена для использования на "больших" вычислительных машинах. Архитектура системы основана на предложениях DataBaseTaskGroup (DBTG), ConferenceonDataSystemsLanguages (CODASYL), организации, ответственной за определение стандартов языка программирования Кобол.

Среди достоинств систем управления данными, основанных на иерархической или сетевой моделях, могут быть названы их компактность и, как правило, высокое быстродействие, а среди недостатков - неуниверсальность, высокая степень зависимости от конкретных данных.

Тема 5. Проектирование БД методом нормализации

При проектировании базы данных решаются две основных проблемы:

- Каким образом отобразить объекты предметной области в абстрактные объекты модели данных? Это логическое проектирование баз данных.
- Как обеспечить эффективность выполнения запросов к базе данных? Это проблема физического проектирования баз данных.

В случае реляционных баз данных трудно представить какие-либо общие рецепты по части физического проектирования. Здесь слишком много зависит от используемой СУБД. Поэтому мы ограничимся вопросами логического проектирования реляционных баз данных, которые существенны при использовании любой реляционной СУБД.

Будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том,

- из каких отношений должна состоять БД и
- какие атрибуты должны быть у этих отношений
- определение первичных, внешних ключей и ограничений целостности.

Проектирование реляционных баз данных с использованием нормализации.

Сначала рассмотрим классический подход, при котором весь процесс проектирования производится в терминах реляционной модели данных, путем последовательных приближений к удовлетворительному набору схем отношений.

Предметная область - это часть реального мира, данные о которой мы хотим отразить в базе данных. Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т.д. Предметная область бесконечна и содержит как существенно важные понятия и данные, так и малозначащие или вообще не значащие данные. Так, если в качестве предметной области выбрать учет товаров на складе, то понятия "накладная" и "счет-фактура" являются существенно важными понятиями, а то, что сотрудница, принимающая накладные, имеет двоих детей - это для учета товаров неважно. Однако, с точки зрения отдела кадров данные о наличии детей являются существенно важными. Таким образом, важность данных зависит от выбора предметной области.

Исходной точкой проектирования является представление предметной области в виде одного или нескольких отношений. На каждом шаге проектирования производится некоторый набор схем отношений, обладающих лучшими свойствами. Процесс проектирования представляет собой процесс нормализации

схем отношений, причем каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

Рассмотрим в качестве предметной области некоторую организацию, выполняющую некоторые проекты. Модель предметной области опишем следующим неформальным текстом:

- Сотрудники организации выполняют проекты.
- Проекты состоят из нескольких заданий.
- Каждый сотрудник может участвовать в одном или нескольких проектах, или временно не участвовать ни в каких проектах.
- Над каждым проектом может работать несколько сотрудников, или временно проект может быть приостановлен, тогда над ним не работает ни один сотрудник.
- Над каждым заданием в проекте работает ровно один сотрудник.
- Каждый сотрудник числится в одном отделе.
- Каждый сотрудник имеет телефон, находящийся в отделе сотрудника.

В ходе дополнительного уточнения того, какие данные необходимо учитывать, выяснилось следующее:

- О каждом сотруднике необходимо хранить табельный номер и фамилию. Табельный номер является уникальным для каждого сотрудника.
- Каждый отдел имеет уникальный номер.
- Каждый проект имеет номер и наименование. Номер проекта является уникальным.
- Каждая работа из проекта имеет номер, уникальный в пределах проекта. Работы в разных проектах могут иметь одинаковые номера.

Первая нормальная форма (1НФ) - это обычное отношение. Любое отношение автоматически уже находится в 1НФ. Напомним кратко свойства отношений (это и будут свойства 1НФ):

- В отношении нет одинаковых кортежей.
- Кортежи не упорядочены.
- Атрибуты не упорядочены и различаются по наименованию.
- Все значения атрибутов атомарны.

В ходе логического моделирования на первом шаге предложено хранить данные в одном отношении, имеющем следующие атрибуты:

СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ (Н_СОТР, ФАМ, Н_ОТД, ТЕЛ, Н_ПРО, ПРОЕКТ, Н_ЗАДАН)
где
Н_СОТР *- табельный номер сотрудника*
ФАМ *- фамилия сотрудника*
Н_ОТД *- номер отдела, в котором числится сотрудник*
ТЕЛ *- телефон сотрудника*
Н_ПРО *- номер проекта, над которым работает сотрудник*
ПРОЕКТ *- наименование проекта, над которым работает сотрудник*
Н_ЗАДАН *- номер задания, над которым работает сотрудник*

*Т.к. каждый сотрудник в каждом проекте выполняет ровно одно задание, то в качестве потенциального первичного ключа отношения необходимо взять пару атрибутов {**Н_СОТР**, **Н_ПРО**}.*
В текущий момент состояние предметной области отражается следующими фактами(выделены ключевые атрибуты):

Н_СОТР	***ФАМ***	***Н_ОТД***	***ТЕЛ***	***Н_ПРО***	***ПРОЕКТ***	***Н_ЗАДАН***
1	*Иванов*	*1*	*11-22-33*	***1***	*Космос*	*1*
1	*Иванов*	*1*	*11-22-33*	***2***	*Климат*	*1*
2	*Петров*	*1*	*11-22-33*	***1***	*Космос*	*2*
3	*Сидоров*	*2*	*33-22-*	***1***	*Космос*	*3*

			11			
3	*Сидоров*	*2*	*33-22-11*	***2***	*Климат*	*2*

Таблица 1 Отношение СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ

Недостатки этого отношения:

- Невозможность вставить сотрудника, который не принимает участие ни в одном проекте, аналогично невозможно вставить проект, если им пока никто не занимается – аномалия вставки. Причина этой аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и о проектах, и о работах по проекту).
- Изменение информации о фамилии, названии проекта, телефоне приводит к необходимости изменять информацию в нескольких местах – аномалия обновления. Причина этой аномалии –избыточность хранения информации (повтор фамилий, названий проектов, телефонов).
- При удалении информации, например о проекте Космос, будет утеряна информация о Петрове, т.к. он работает только в том проекте – аномалия удаления. Причина этой аномалии - хранение в одном отношении разнородной информации.

*Вывод - Отношение **СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ** находится в 1НФ, но при этом, как было показано выше, логическая модель данных не соответствует модели предметной области. База данных, основанная на такой модели, будет работать неправильно. Таким образом, первой нормальной формы недостаточно для правильного моделирования данных.*

Для устранения указанных аномалий, для правильного проектирования модели данных применяется метод нормализации отношений.

Перевод отношения в следующую нормальную форму осуществляется методом "декомпозиции без потерь". Такая декомпозиция должна обеспечить то, что запросы (выборка данных по условию) к исходному отношению и к отношениям, получаемым в результате декомпозиции, дадут одинаковый результат.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1NF);
- вторая нормальная форма (2NF);
- третья нормальная форма (3NF);
- нормальная форма Бойса-Кодда (BCNF);
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).

Основные свойства нормальных форм:

- каждая следующая нормальная форма «лучше» предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих форм сохраняются.

В основе процесса проектирования лежит метод нормализации или декомпозиции отношения, находящегося в предыдущей нормальной форме, в два или более отношения, удовлетворяющих требованиям следующей нормальной формы.

Метод нормализации (а значит процесс проектирования) основан на понятии функциональной зависимости атрибутов отношения – фундаментальное понятие в теории реляционных баз данных.

Определение. Функциональная зависимость

В отношении R атрибут Y функционально зависит от атрибута X (X и Y могут быть составными) в том и только в том случае, если каждому значению X соответствует в точности одно значение Y: R.X (r) R.Y $X \rightarrow Y$.

***Пример 1**. В отношении **СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ** можно привести следующие примеры функциональных зависимостей:*

Зависимость атрибутов от ключа отношения:

{Н_СОТР, Н_ПРО} $\rightarrow$ ***ФАМ***
{Н_СОТР, Н_ПРО} $\rightarrow$ ***Н_ОТД***
{Н_СОТР, Н_ПРО} $\rightarrow$ ***ТЕЛ***
{Н_СОТР, Н_ПРО} $\rightarrow$ ***ПРОЕКТ***
{Н_СОТР, Н_ПРО} $\rightarrow$ ***Н_ЗАДАН***

Зависимость атрибутов, характеризующих сотрудника от табельного номера сотрудника:

Н_СОТР $\rightarrow$ ***ФАМ***
Н_СОТР $\rightarrow$ ***Н_ОТД***
Н_СОТР $\rightarrow$ ***ТЕЛ***

Зависимость наименования проекта от номера проекта:

Н_ПРО $\rightarrow$ ***ПРОЕКТ***
Зависимость номера телефона от номера отдела:
Н_ОТД $\rightarrow$ ***ТЕЛ***
Замечание. Если атрибуты X составляют потенциальный ключ отношения R, то любой атрибут отношения функционально зависит от X.

Необходимо отличать понятия функциональной зависимости в математике и РБД. В математическом понятии функции для фиксированного значения X соответствующее значение функции $y = f(x)$ *всегда одно и то же. В противоположность этому в отношениях значение зависимого атрибута может принимать различные значения в различных состояниях базы данных. Например, атрибут* ***ФАМ*** *функционально зависит от атрибута* ***Н_СОТР****. Предположим, что сейчас сотрудник с табельным номером 1 имеет фамилию «Иванов», но сотрудник может сменить фамилию, например на «Сидоров».*

Определение. Неключевой атрибут

Неключевым атрибутом называется любой атрибут отношения, не входящий в состав любого ключа (в частности первичного ключа).

Пример ФАМ, Н_ОТД, ТЕЛ, ПРОЕКТ, Н_ЗАДАН

Вторая нормальная форма

Определение. Отношение R находится во второй нормальной форме (2NF) в том и только в том случае, когда находится в 1NF, и каждый неключевой атрибут полностью зависит от первичного ключа (т.е. от всех атрибутов в него входящих), если первичный ключ отношения является простым (т.е. ключ состоит из одного атрибута), то отношение автоматически находится в 2НФ.

Отношение ***СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ*** *не находится в 2НФ, т.к. есть атрибуты, зависящие от части составного первичного ключа:*
Зависимость атрибутов, характеризующих сотрудника от табельного номера сотрудника является зависимостью от части сложного ключа:
Н_СОТР $\rightarrow$ ***ФАМ*** ***Н_СОТР*** $\rightarrow$ ***Н_ОТД*** ***Н_СОТР*** $\rightarrow$ ***ТЕЛ***

Зависимость наименования проекта от номера проекта является зависимостью от части первичног ключа: ***Н_ПРО*** $\rightarrow$ ***ПРОЕКТ***
Для того, чтобы устранить зависимость атрибутов от части первичного ключа, нужно произвести ***декомпозицию*** *отношения на несколько отношений. При этом те атрибуты, которые зависят от части первичного ключа, выносятся в отдельное отношение.*
Отношение ***СОТРУДНИКИ_ОТДЕЛЫ_ПРОЕКТЫ*** *декомпозируем на три отношения –*

СОТРУДНИКИ_ОТДЕЛЫ, ПРОЕКТЫ, ЗАДАНИЯ.

Отношение ***СОТРУДНИКИ_ОТДЕЛЫ (Н_СОТР, ФАМ, Н_ОТД, ТЕЛ):***
Функциональные зависимости:
Зависимость атрибутов, характеризующих сотрудника от табельного номера сотрудника:
Н_СОТР $\rightarrow$ ***ФАМ***
Н_СОТР $\rightarrow$ ***Н_ОТД***
Н_СОТР $\rightarrow$ ***ТЕЛ***
Зависимость номера телефона от номера отдела:
Н_ОТД $\rightarrow$ ***ТЕЛ***

Н_СОТР	ФАМ	Н_ОТД	ТЕЛ
1	Иванов	1	11-22-33
2	Петров	1	11-22-33
3	Сидоров	2	33-22-11

Отношение СОТРУДНИКИ_ОТДЕЛЫ

Отношение ***ПРОЕКТЫ (Н_ПРО, ПРОЕКТ):***
Функциональные зависимости:
Н_ПРО $\rightarrow$ ***ПРОЕКТ***

Н_ПРО	ПРОЕКТ

1	*Космос*
2	*Климат*

Отношение ПРОЕКТЫ

*Отношение **ЗАДАНИЯ (Н_СОТР, Н_ПРО, Н_ЗАДАН)**:*
Функциональные зависимости:
{Н_СОТР, Н_ПРО}** $\rightarrow$ **Н_ЗАДАН

Н_СОТР	***Н_ПРО***	***Н_ЗАДАН***
1	*1*	*1*
1	*2*	*1*
2	*1*	*2*
3	*1*	*3*
3	*2*	*2*

Отношения ЗАДАНИЯ

Анализ декомпозированных отношений:

*Отношения, полученные в результате декомпозиции, находятся в 2НФ. Действительно, отношения **СОТРУДНИКИ_ОТДЕЛЫ** и **ПРОЕКТЫ** имеют простые ключи, следовательно автоматически находятся в 2НФ, отношение **ЗАДАНИЯ** имеет сложный ключ, но единственный неключевой атрибут **Н_ЗАДАН** функционально зависит от всего ключа **{Н_СОТР, Н_ПРО}**.*

Часть аномалий устранена:

- *Так, данные о сотрудниках и проектах теперь хранятся в различных отношениях, поэтому при появлении сотрудников, не участвующих ни в одном проекте просто добавляются кортежи в отношение **СОТРУДНИКИ_ОТДЕЛЫ**. Точно также, при появлении проекта, над которым не работает ни один сотрудник, просто вставляется кортеж в отношение **ПРОЕКТЫ**.*
- *Фамилии сотрудников и наименования проектов теперь хранятся без избыточности. Если сотрудник сменит фамилию или проект сменит наименование, то такое обновление будет произведено в одном месте*
- *Если по проекту временно прекращены работы, но требуется, чтобы сам проект сохранился, то для этого проекта удаляются соответствующие кортежи в отношении **ЗАДАНИЯ**, а данные о самом проекте и данные о сотрудниках, участвовавших в проекте, остаются в отношениях **ПРОЕКТЫ** и **СОТРУДНИКИ_ОТДЕЛЫ**.*

Тем не менее, часть аномалий разрешить не удалось:

- *В отношение **СОТРУДНИКИ_ОТДЕЛЫ** нельзя вставить кортеж (4, Пушников, 1, 33-22-11), т.к. при этом получится, что два сотрудника из 1-го отдела (Иванов и Пушников) имеют разные номера телефонов, а это противоречит модели предметной области. В этой ситуации можно предложить два решения, в зависимости от того, что реально произошло в предметной области. Другой номер телефона может быть введен по двум причинам - по ошибке человека, вводящего данные о новом сотруднике, или потому что номер в отделе действительно изменился. Тогда можно написать триггер, который при вставке записи о сотруднике проверяет, совпадает ли телефон с уже имеющимся телефоном у другого сотрудника этого же отдела. Если номера отличаются, то система должна задать вопрос, оставить ли старый номер в отделе или заменить его новым. Если нужно оставить старый номер (новый номер введен ошибочно), то кортеж с данными о новом сотруднике будет вставлен, но номер телефона будет у него будет тот, который уже есть в отделе (в данном случае, 11-22-33). Если же номер в отделе действительно изменился, то кортеж будет вставлен с новым номером, и одновременно будут изменены номера телефонов у всех сотрудников этого же отдела. И в том и в другом случае не обойтись без разработки громоздкого триггера.).*
- *если в отделе меняется номер телефона, то такие изменения необходимо одновременно выполнить во всех местах, где этот номер телефона встречаются, иначе отношение станет некорректным – **аномалия обновления**. Причина аномалии - избыточность данных, также порожденная тем, что в одном отношении хранится разнородная информация.*
- *При удалении некоторых данных по-прежнему может произойти потеря другой информации. Например, если удалить сотрудника Сидорова, то будет потеряна информация о том, что в отделе номер 2 находится телефон 33-22-11 – **аномалия удаления**. Причина аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и об отделах).*

Вывод - логическая модель данных не соответствует предметной области. База данных, основанная на такой модели, будет работать неправильно. Хотя основные трудности теперь связаны только с одним отношением ***СОТРУДНИКИ_ОТДЕЛЫ****.*

Третья нормальная форма

Определение. Атрибуты называются взаимно независимыми, если ни один из них не является функционально зависимым от другого.

Определение. Отношение R находится в третьей нормальной форме (3НФ) тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимно независимы.

Отношение ***СОТРУДНИКИ_ОТДЕЛЫ*** *не находится в 3НФ, т.к. имеется функциональная зависимость неключевых атрибутов (зависимость номера телефона от номера отдела):*

Н_ОТД $\rightarrow$ ***ТЕЛ***

Для того, чтобы устранить зависимость неключевых атрибутов, нужно произвести декомпозицию отношения на несколько отношений. При этом те неключевые атрибуты, которые являются зависимыми, выносятся в отдельное отношение.

Отношение ***СОТРУДНИКИ_ОТДЕЛЫ*** *декомпозируем на два отношения -* ***СОТРУДНИКИ****,* ***ОТДЕЛЫ****.*

Отношение ***СОТРУДНИКИ (Н_СОТР, ФАМ, Н_ОТД):***

Функциональные зависимости:

Зависимость атрибутов, характеризующих сотрудника от табельного номера сотрудника:

Н_СОТР $\rightarrow$ ***ФАМ***

Н_СОТР $\rightarrow$ ***Н_ОТД***

Н_СОТР	***ФАМ***	***Н_ОТД***
1	*Иванов*	*1*
2	*Петров*	*1*
3	*Сидоров*	*2*

Отношение СОТРУДНИКИ

Отношение ***ОТДЕЛЫ (Н_ОТД, ТЕЛ):***

Функциональные зависимости:

Зависимость номера телефона от номера отдела:

Н_ОТД $\rightarrow$ ***ТЕЛ***

Н_ОТД	***ТЕЛ***
1	*11-22-33*
2	*33-22-11*

Отношение ОТДЕЛЫ

Обратим внимание на то, что атрибут ***Н_ОТД****, не являвшийся ключевым в отношении* ***СОТРУДНИКИ_ОТДЕЛЫ****, становится потенциальным ключом в отношении* ***ОТДЕЛЫ****. Именно за счет этого устраняется избыточность, связанная с многократным хранением одних и тех же номеров телефонов.*

Вывод. Таким образом, все обнаруженные аномалии обновления устранены. Реляционная модель, состоящая из четырех отношений ***СОТРУДНИКИ****,* ***ОТДЕЛЫ****,* ***ПРОЕКТЫ****,* ***ЗАДАНИЯ****, находящихся в третьей нормальной форме соответствует описанной предметной области.*

На практике третья нормальная форма схем отношений достаточна в большинстве случаев, и приведением к третьей нормальной форме процесс проектирования реляционной базы данных обычно заканчивается. Однако иногда полезно продолжить процесс нормализации.

Замечание. На практике, при создании логической модели данных, как правило, не следуют прямо приведенному алгоритму нормализации. Опытные разработчики обычно сразу строят отношения в 3НФ. Тем не менее, приведенный алгоритм важен:

- этот алгоритм показывает, какие проблемы возникают при разработке слабо нормализованных отношений.
- как правило, модель предметной области никогда не бывает правильно разработана с первого шага. Эксперты предметной области могут забыть о чем-либо упомянуть, разработчик может неправильно понять эксперта, во время разработки могут измениться правила, принятые в предметной области, и т.д. Все это может привести к появлению новых зависимостей, которые отсутствовали в первоначальной модели

предметной области. Тут как раз и необходимо использовать алгоритм нормализации хотя бы для того, чтобы убедиться, что отношения остались в 3НФ, и логическая модель не ухудшилась.

Что получилось – обобщение примера:

Н_СОТР	***ФАМ***	***Н_ОТД***
1	*Иванов*	*1*
2	*Петров*	*1*
3	*Сидоров*	*2*

Отношение СОТРУДНИКИ

Н_ОТД	***ТЕЛ***
1	*11-22-33*
2	*33-22-11*

Отношение ОТДЕЛЫ

Н_ПРО	***ПРОЕКТ***
1	*Космос*
2	*Климат*

Отношение ПРОЕКТЫ

Н_СОТР	***Н_ПРО***	***Н_ЗАДАН***
1	*1*	*1*
1	*2*	*1*
2	*1*	*2*
3	*1*	*3*
3	*2*	*2*

Отношения ЗАДАНИЯ

НФБК (Нормальная Форма Бойса-Кодда)

При приведении отношений при помощи алгоритма нормализации к отношениям в 3НФ неявно предполагалось, что все отношения содержат один потенциальный ключ. Это не всегда верно. Рассмотрим следующий пример отношения, содержащего два ключа.

Пример 1. Пусть требуется хранить данные о поставках деталей некоторыми поставщиками. Предположим, что наименования поставщиков являются уникальными. Кроме того, каждый поставщик имеет свой уникальный номер. Данные о поставках можно хранить в следующем отношении:

Номер поставщика PNUM	**Наименование поставщика PNAME**	**Номер детали DNUM**	**Поставляемое количество VOLUME**
1	Фирма 1	1	100
1	Фирма 1	2	200
1	Фирма 1	3	300
2	Фирма 2	1	150
2	Фирма 2	2	250
3	Фирма 3	1	1000

Отношение "Поставки"

Данное отношение содержит два потенциальных ключа - **{*PNUM, DNUM*}** и **{*PNAME, DNUM*}**. Видно, что данные хранятся в отношении с избыточностью - при изменении наименования поставщика, это наименование нужно изменить во всех кортежах, где оно встречается.

Выявляем имеющиеся функциональные зависимости (как обычно, курсивом выделены ключевые атрибуты):

PNUM $\rightarrow$ ***PNAME***.

PNAME $\rightarrow$ ***PNUM***

{PNUM, DNUM} $\rightarrow$**VOLUME**
{PNUM, DNUM} $\rightarrow$***PNAME***

{PNAME, DNUM} $\rightarrow$**VOLUME**
{PNAME, DNUM} $\rightarrow$***PNUM***

Данное отношение не содержит *неключевых* атрибутов, зависящих от части сложного ключа (см. определение 2НФ). Действительно, от части сложного ключа зависят атрибуты ***PNAME*** и ***PNUM***, но они сами являются ключевыми. Таким образом, отношение находится в 2НФ.

Кроме того, отношение не содержит зависимых друг от друга *неключевых* атрибутов, т.к. неключевой атрибут всего один - **VOLUME** (см. определение 3НФ). Таким образом, показано, что отношение "Поставки" находится в 3НФ.

Таким образом, описанный ранее алгоритм нормализации неприменим к данному отношению. Очевидно, однако, что аномалия данного отношения устраняется путем декомпозиции его на два следующих отношения:

Номер поставщика PNUM	**Наименование поставщика PNAME**
1	**Фирма 1**
2	**Фирма 2**
3	**Фирма 3**

Отношение "Поставщики"

Номер поставщика PNUM	**Номер детали DNUM**	**Поставляемое количество VOLUME**
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	1	1000

Отношение "Поставки-2"

Определение . Отношение R находится в нормальной форме Бойса-Кодда (НФБК) тогда и только тогда, когда множество атрибутов X (детерминанты) всех функциональных зависимостей являются потенциальными ключами.

X $\rightarrow$ Y X –множество атрибутов X – детерминанта; Y – зависимая часть

Отношение "Поставки" не находится в НФБК, т.к. имеются зависимости (***PNUM*** $\rightarrow$***PNAME*** и ***PNAME*** $\rightarrow$***PNUM***), детерминанты которых не являются потенциальными ключами.

Для того чтобы устранить зависимость от детерминантов, не являющихся потенциальными ключами, необходимо провести декомпозицию, вынося эти детерминанты и зависимые от них части в отдельное отношение. Отношения "Поставщики" и "Поставки-2", полученные в результате декомпозиции находятся в НФБК.

Замечание. Отношение "Поставки-2", полученное в результате декомпозиции имеет всего *один потенциальный ключ*. Поэтому, для анализа отношения "Поставки-2" не требуется привлекать определение НФБК, достаточно определения 3НФ.

Отношение "Поставщики" имеет два потенциальных ключа, но, т.к. других атрибутов в нем нет, оно уже так просто устроено, что упростить его дальше нельзя.

4НФ (Четвертая Нормальная Форма)

Рассмотрим следующий пример. Пусть требуется учитывать данные об абитуриентах, поступающих в ВУЗ. При анализе предметной области были выделены следующие требования:

- Каждый абитуриент имеет право сдавать экзамены на несколько факультетов одновременно.
- Каждый факультет имеет свой список сдаваемых предметов.
- Один и тот же предмет может сдаваться на нескольких факультетах.
- Абитуриент обязан сдавать все предметы, указанные для факультета, на который он поступает, несмотря на то, что он, может быть, уже сдавал такие же предметы на другом факультете.

Предположим, что нам требуется хранить данные о том, какие предметы должен сдавать каждый абитуриент. Попытаемся хранить данные в одном отношении "Абитуриенты-Факультеты-Предметы":

Абитуриент	Факультет	Предмет
Иванов	Математический	Математика
Иванов	Математический	Информатика
Иванов	Физический	Математика
Иванов	Физический	Физика
Петров	Математический	Математика
Петров	Математический	Информатика

Отношение "Абитуриенты-Факультеты-Предметы"

В данный момент в отношении хранится информация о том, что абитуриент Иванов поступает на два факультета (математически и физический), а абитуриент Петров - только на математический. Кроме того, можно сделать вывод, что на математическом факультете нужно сдавать математику и информатику, а на физическом - математику и физику.

Кажется, что в отношении имеется аномалия обновления, связанная с тем, что дублируются фамилии абитуриентов, наименования факультетов и наименования предметов. Однако эта аномалия легко устраняется стандартным способом - вынесением всех наименований в отдельные отношения, оставляя в исходном отношении только соответствующие номера:

Номер Абитуриента	Номер Факультета	Номер Предмета
1	1	1
1	1	2
1	2	1
1	2	3
2	1	1
2	1	2

Модифицированное отношение "Абитуриенты-Факультеты-Предметы"

Номер Абитуриента	Абитуриент
1	Иванов
2	Петров

Отношение "Абитуриенты"

Номер Факультета	Факультет
1	Математический
2	Физический

Отношение "Факультеты"

Номер Предмета	Предмет
1	Математика
2	Информатика
3	Физика

Отношение "Предметы"

Теперь каждое наименование встречается только в одном месте.

И все-таки как в исходном, так и в модифицированном отношении имеются аномалии обновления, возникающие при попытке вставить или удалить кортежи.

Аномалия вставки. При попытке добавить в отношение "Абитуриенты-Факультеты-Предметы" новый кортеж, например (Сидоров, Математический, Математика), мы обязаны добавить также и кортеж

(Сидоров, Математический, Информатика), т.к. все абитуриенты математического факультета обязаны иметь один и тот же список сдаваемых предметов. Соответственно, при попытке вставить в модифицированное отношении кортеж (3, 1, 1), мы обязаны вставить в него также и кортеж (3, 1, 2).

Аномалия удаления. При попытке удалить кортеж (Иванов, Математический, Математика), мы обязаны удалить также и кортеж (Иванов, Математический, Информатика) по той же самой причине.

Таким образом, вставка и удаление кортежей не может быть выполнена независимо от других кортежей отношения.

Кроме того, если мы удалим кортеж (Иванов, Физический, Математика), а вместе с ним и кортеж (Иванов, Физический, Физика), то будет потеряна информация о предметах, которые должны сдаваться на физическом факультете.

Декомпозиция отношения "Абитуриенты-Факультеты-Предметы" для устранения указанных аномалий не может быть выполнена на основе функциональных зависимостей, т.к. это отношение не содержит никаких функциональных зависимостей. Это отношение является полностью ключевым, т.е. ключом отношения является все множество атрибутов. Но ясно, что какая-то взаимосвязь между атрибутами имеется. Эта взаимосвязь описывается понятием многозначной зависимости.

В отношении "Абитуриенты-Факультеты-Предметы" имеется многозначная зависимость Факультет $\rightarrow\rightarrow$ Абитуриент|Предмет.

Словами это можно выразить так - для каждого факультета (для каждого значения из X) каждый поступающий на него абитуриент (значение из Y) сдает *один и тот же список* предметов (набор значений из Z), и для каждого факультета (для каждого значения из X) каждый сдаваемый на факультете экзамен (значение из Z) сдается *одним и тем же списком* абитуриентов (набор значений из Y). Именно наличие этой зависимости не позволяет независимо вставлять и удалять кортежи. Кортежи обязаны вставляться и удаляться одновременно *целыми наборами*.

В отношении "Абитуриенты-Факультеты-Предметы" имеется именно нетривиальная многозначная зависимость Факультет $\rightarrow\rightarrow$ Абитуриент|Предмет.

Определение . Отношение R находится в ***четвертой нормальной форме*** (***4НФ***) тогда и только тогда, когда отношение находится в НФБК и *не содержит нетривиальных многозначных зависимостей*.

Отношение "Абитуриенты-Факультеты-Предметы" находится в НФБК, но не в 4НФ. Это отношение можно без потерь декомпозировать на отношения:

Факультет	**Абитуриент**
Математический	Иванов
Физический	Иванов
Математический	Петров

Отношение "Факультеты-Абитуриенты"

Факультет	**Предмет**
Математический	Математика
Математический	Информатика
Физический	Математика
Физический	Физика

Отношение "Факультеты-Предметы"

В полученных отношениях устранены аномалии вставки и удаления, характерные для отношения "Абитуриенты-Факультеты-Предметы".

Тема 6. Проектирование БД с использованием семантических моделей

Широкое распространение реляционных СУБД и их использование в самых разнообразных приложениях показывает, что реляционная модель данных достаточна для моделирования разнообразных предметных областей. Однако проектирование реляционной базы данных в терминах отношений на основе кратко рассмотренного нами в двух предыдущих лекциях механизма нормализации часто представляет собой очень сложный и неудобный для проектировщика процесс.

Ограниченность реляционной модели при проектировании баз данных

При использовании в проектировании ограниченность реляционной модели проявляется в следующих аспектах.

- Модель не обеспечивает достаточных средств для представления смысла данных. Семантика реальной предметной области должна независимым от модели способом представляться в голове проектировщика. В частности, это относится к отмечавшейся нами ранее проблеме представления ограничений целостности, выходящих за пределы ограничений первичного и внешнего ключа.
- Во многих прикладных областях трудно моделировать предметную область на основе плоских таблиц40). В ряде случаев на самой начальной стадии проектирования дизайнеру приходится нелегко, поскольку от него требуется описать предметную область в виде одной (возможно, даже ненормализованной) таблицы.
- Хотя весь процесс проектирования происходит на основе учета зависимостей, реляционная модель не предоставляет какие-либо формализованные средства для представления этих зависимостей.
- Несмотря на то, что процесс проектирования начинается с выделения некоторых существенных для приложения объектов предметной области («сущностей») и выявления связей между этими сущностями, реляционная модель данных не предлагает какого-либо механизма для разделения сущностей и связей.

Семантические модели данных

Потребность проектировщиков баз данных в более удобных и мощных средствах моделирования предметной области вызвала к жизни направление семантических моделей данных. Хотя любая развитая семантическая модель данных, как и реляционная модель, включает структурную, манипуляционную и целостную части, главным назначением семантических моделей является обеспечение возможности выражения семантики данных.

Прежде чем мы коротко рассмотрим особенности двух распространенных семантических моделей, остановимся на возможных областях их применения. Чаще всего на практике семантическое моделирование используется на первой стадии проектирования базы данных. При этом в терминах семантической модели производится концептуальная схема базы данных, которая затем вручную преобразуется к реляционной (или какой-либо другой) схеме. Этот процесс выполняется под управлением методик, в которых достаточно четко оговорены все этапы такого преобразования. Основным достоинством данного подхода является отсутствие потребности в дополнительных программных средствах, поддерживающих семантическое моделирование. Требуется только знание основ выбранной семантической модели и правил преобразования концептуальной схемы в реляционную схему.

Следует заметить, что многие начинающие проектировщики баз данных недооценивают важность семантического моделирования вручную. Зачастую это воспринимается как дополнительная и излишняя работа. Эта точка зрения абсолютно неверна. Во-первых, построение мощной и наглядной концептуальной схемы БД позволяет более полно оценить специфику моделируемой предметной области и избежать возможных ошибок на стадии проектирования схемы реляционной БД. Во-вторых, на этапе семантического моделирования производится важная документация (хотя бы в виде вручную нарисованных диаграмм и комментариев к ним), которая может оказаться очень полезной не только при проектировании схемы реляционной БД, но и при эксплуатации, сопровождении и развитии уже заполненной БД.

Неоднократно приходилось и приходится наблюдать ситуации, в которых отсутствие такого рода документации существенно затрудняет внесение даже небольших изменений в схему существующей реляционной БД. Конечно, это относится к случаям, когда проектируемая БД содержит не слишком малое число таблиц. Скорее всего, без семантического моделирования можно обойтись, если число таблиц не превышает десяти, но оно совершенно необходимо, если БД включает более сотни таблиц. Для справедливости заметим, что процедура создания концептуальной схемы вручную с ее последующим преобразованием в реляционную схему БД затруднительна в случае больших БД (содержащих несколько сотен таблиц). Причины, по всей видимости, не требуют пояснений.

История систем автоматизации проектирования баз данных (CASE-средств) началась с автоматизации процесса рисования диаграмм, проверки их формальной корректности, обеспечения средств долговременного хранения диаграмм и другой проектной документации. Конечно, компьютерная поддержка работы с диаграммами для проектировщика БД очень полезна. Наличие электронного архива проектной документации помогает при эксплуатации, администрировании и сопровождении базы данных. Но система, которая ограничивается поддержкой рисования диаграмм, проверкой их корректности и хранением, напоминает текстовый редактор, поддерживающий ввод, редактирование и проверку синтаксической корректности конструкций некоторого языка программирования, но существующий отдельно от компилятора. Кажется естественным желание расширить такой редактор функциями компилятора, и это действительно возможно, поскольку известна техника компиляции конструкций языка программирования в коды целевого компьютера. Но коль скоро имеется четкая методика преобразования концептуальной схемы БД в реляционную схему, то почему бы не выполнить программную реализацию соответствующего «компилятора» и не включить ее в состав системы проектирования баз данных?

Эта идея, естественно, показалась разумной производителям CASE-средств проектирования БД. Подавляющее большинство подобных систем, представленных на рынке, обеспечивает автоматизированное преобразование диаграммных концептуальных схем баз данных, представленных в той или иной

семантической модели данных, в реляционные схемы, специфицированные чаще всего на языке SQL. У читателя может возникнуть вопрос, почему в предыдущем предложении говорится про «автоматизированное», а не про «автоматическое» преобразование? Все дело в том, что в типичной схеме SQL-ориентированной БД могут содержаться определения многих объектов (ограничений целостности общего вида, триггеров и хранимых процедур и т. д.), которые невозможно сгенерировать автоматически на основе концептуальной схемы. Поэтому на завершающем этапе проектирования реляционной схемы снова требуется ручная работа проектировщика.

Еще раз обратите внимание на то, какой ход рассуждений привел нас к выводу о возможности автоматизации процесса преобразования концептуальной схемы БД в реляционную схему. Если создатели семантической модели данных предоставляют методику преобразования концептуальных схем в реляционные, то почему бы не реализовать программу, которая производит те же преобразования, следуя той же методике? Зададимся теперь другим, но, по существу, схожим вопросом. Если создатели семантической модели данных предоставляют язык (например, диаграммный), используя который проектировщики БД на основе исходной информации о предметной области могут сформировать концептуальную схему БД, то почему бы не реализовать программу, которая сама генерирует концептуальную схему БД в соответствующей семантической модели, используя исходную информацию о предметной области? Хотя нам не известны коммерческие CASE-средства проектирования БД, поддерживающие такой подход, экспериментальные системы успешно существовали. Они представляли собой интегрированные системы проектирования с автоматизированным созданием концептуальной схемы на основе интервью с экспертами предметной области и последующим преобразованием концептуальной схемы в реляционную.

Как правило, CASE-средства, автоматизирующие преобразование концептуальной схемы БД в реляционную, производят реляционную схему базы данных в третьей нормальной форме. Нормализация более высокого уровня усложняет программную реализацию и редко требуется на практике.

Наконец, третья возможность, которую следует упомянуть, хотя она еще не вышла (или только выходит, а может быть, так никогда и не выйдет) за пределы исследовательских и экспериментальных проектов, – это работа с базой данных в семантической модели, т.е. СУБД, основанные на семантических моделях данных. При этом снова рассматриваются два варианта: обеспечение пользовательского интерфейса на основе семантической модели данных с автоматическим отображением конструкций этого интерфейса в реляционную модель данных (это задача примерно того же уровня сложности, что и автоматическая компиляция концептуальной схемы базы данных в реляционную схему) и прямая реализация СУБД, основанная на какой-либо семантической модели данных. Многие авторитетные специалисты полагают, что ближе всего ко второму подходу объектно-ориентированные СУБД, чьи модели данных по многим параметрам близки к семантическим моделям (хотя в некоторых аспектах они более мощны, а в некоторых – более слабы).

Далее переходим к использованию терминов таблица, строка и столбец вместо строгих реляционных терминов отношение, атрибут и таблица, поскольку здесь под «реляционными» базами данных понимаются, главным образом, SQL-ориентированные базы данных, для которых эта упрощенная терминология более естественна.

Многие сторонники реляционного подхода считают отсутствие раздельного представления сущностей и связей преимуществом реляционной модели данных, мотивируя это тем, что зачастую то, что вчера считалось сущностью, сегодня разумнее принять за связь, и наоборот. Это, безусловно, верно с точки зрения поддержки и модификации существующих реляционных баз данных, но отнюдь не так с точки зрения проектирования базы данных.

Издавна существует отдельный класс программных систем, предназначенных для автоматизации проектирования новых продуктов в разных областях промышленности – автомобилестроении, аэрокосмической промышленности, электронной промышленности и т.д. Очевидно, что процесс проектирования автомобиля принципиально отличается от процесса проектирования микропроцессора, но, тем не менее, для обозначения любой Системы Автоматизации Проектирования используется собирательный термин САПР (CAD – Computer Aided Design). Это оправдывается тем, что разные подклассы САПР имеют гораздо больше общих черт, чем различий. Так вот, система автоматизации проектирования БД по своему назначению и строению в большей степени является системой класса САПР, чем системой класса CASE (Computer Aided Software Engineering). По всей видимости, средства автоматизированной поддержки проектирования баз данных стали в свое время называть CASE-средствами, поскольку они обычно включали не только инструменты для поддержки проектирования, но и инструменты, поддерживающие проектирование и разработку приложений баз данных. В последние годы такие инструменты все реже производятся в виде одного пакета, и сам термин «CASE-средство» почти вышел из употребления. Тем не менее, поскольку не появилось какое-либо другое собирательное название средств поддержки проектирвания баз данных, мы будем продолжать использовать именно этот термин.

Семантическая модель Entity-Relationship (Сущность-Связь)

В этой лекции мы кратко рассмотрим некоторые черты одной из наиболее популярных семантических моделей данных – модели «Сущность-Связь» (часто ее называют кратко ER-моделью от Entity-Relationship).

Здесь следует сделать два замечания, касающиеся, главным образом, терминологии. Оба термина relation и relationship могут быть переведены на русский язык как отношение. Поэтому в русскоязычной литературе ER-модель иногда называют моделью сущность-отношение, а иногда и реляционной семантической моделью. Наверное, в этом нет ничего страшного, если говорить о ER-модели в отрыве от тематики проектирования реляционных баз данных.

Но если требуется одновременно использовать термины ER-модели и реляционной модели данных, то, безусловно, требуется применять для терминов relation и relationship разные русские эквиваленты. За этими терминами стоят весьма различные понятия. В реляционной модели отношение (relation) – это единственная родовая структура данных. С помощью этого же механизма представляются «связанные» сущности (вспомните, например, про внешние ключи). Как мы увидим немного позже, в ER-модели для представления схемы базы данных используются два равноправных понятия – сущность и связь. Связи в ER-модели играют роль, отличную от той, какую играют отношения в реляционной модели данных.

Кроме того, в русскоязычную терминологию вошла и чистая транслитерация термина relation именно в смысле отношение. Мы говорим, например, про реляционную модель данных, реляционную алгебру и т. д., понимая модель данных, основанную на отношениях, алгебру отношений и т. п. По этому поводу, по крайней мере, в контексте баз данных, разумно окончательно зарезервировать термины relation и отношение для обозначения понятий реляционной модели данных, а для термина relationship использовать другой допустимый русскоязычный эквивалент – связь.

На использовании разных вариантов ER-модели основано большинство современных подходов к проектированию баз данных (главным образом, реляционных). Модель была предложена Питером Ченом (Peter Chen) в 1976г. Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов. Простота и наглядность представления концептуальных схем баз данных в ER-модели привели к ее широкому распространению в CASE-системах, поддерживающих автоматизированное проектирование реляционных баз данных. Среди множества разновидностей ER-моделей одна из наиболее популярных и развитых применялась в системе CASE компании Oracle. Мы обсудим некоторый упрощенный вариант этой модели. Если говорить более точно, сосредоточимся на ее структурной и целостной частях.

Основные понятия ER-модели

Основными понятиями ER-модели являются сущность, связь и атрибут. Сущность – это реальный или представляемый объект, информация о котором должна сохраняться и быть доступной.43) В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности – это имя типа, а не некоторого конкретного экземпляра этого типа. Для большей выразительности и лучшего понимания имя сущности может сопровождаться примерами конкретных экземпляров этого типа.

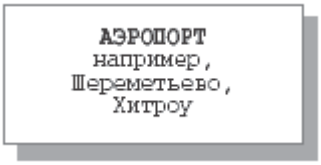

Рис. 5.1. Пример типа сущности

На рис. 5.1 изображена сущность АЭРОПОРТ с примерными экземплярами «Шереметьево» и «Хитроу». Эта примитивная диаграмма тем не менее несет важную информацию. Во-первых, она показывает, что в базе данных будут содержаться однотипные структуры данных (экземпляры сущности), описывающие аэропорты. Во-вторых, поскольку в жизни существует несколько точек зрения на аэропорты (например, точка зрения пилота, точка зрения пассажира, точка зрения администратора) и этим точкам зрения соответствуют разные структуры данных, то приведенные примеры аэропортов позволяют несколько сузить допустимый набор точек зрения. В нашем случае приведены примеры международных аэропортов, так что, скорее всего, имеется точка зрения пассажира или пилота международных авиарейсов.

При определении типа сущности необходимо гарантировать, что каждый экземпляр сущности может быть отличим от любого другого экземпляра той же сущности. Это требование в некотором роде аналогично требованию отсутствия кортежей-дубликатов в реляционных таблицах.

Связь – это графически изображаемая ассоциация, устанавливаемая между двумя типами сущностей. Как и сущность, связь – это типовое понятие, все экземпляры обоих связываемых типов сущностей подчиняются устанавливаемым правилам связывания. Поэтому правильнее говорить о типе связи, устанавливаемой между типами сущности, и об экземплярах типа связи, устанавливаемых между

экземплярами типа сущности.45) В обсуждаемом здесь варианте ER-модели эта ассоциация всегда является бинарной и может существовать между двумя разными типами сущностей или между типом сущности и им же самим (рекурсивная связь). В любой связи выделяются два конца (в соответствии с существующей парой связываемых сущностей), на каждом из которых указываются имя конца связи, степень конца связи (сколько экземпляров данного типа сущности должно присутствовать в каждом экземпляре данного типа связи), обязательность связи (т. е. любой ли экземпляр данного типа сущности должен участвовать в некотором экземпляре данного типа связи).46)

Связь представляется в виде ненаправленной линии, соединяющей две сущности или ведущей от сущности к ней же самой. При этом в месте «стыковки» связи с сущностью используются:

- трехточечный вход в прямоугольник сущности, если для этой сущности в связи могут (или должны) использоваться много (many) экземпляров сущности;

одноточечный вход, если в связи может (или должен) участвовать только один экземпляр сущности.

Обязательный конец связи изображается сплошной линией, а необязательный – прерывистой линией.

Связь между сущностями БИЛЕТ и ПАССАЖИР, показанная на рис. 5.2, связывает билеты и пассажиров. Конец связи с именем «для» позволяет связывать с одним пассажиром более одного билета, причем каждый билет должен быть связан с каким-либо пассажиром. Конец связи с именем «имеет» показывает, что каждый билет может принадлежать только одному пассажиру, причем пассажир не обязан иметь хотя бы один билет.

Рис. 5.2. Пример типа связи

Лаконичная устная трактовка изображенной диаграммы состоит в следующем:

- каждый БИЛЕТ предназначен для одного и только одного ПАССАЖИРА;
- каждый ПАССАЖИР может иметь один или более БИЛЕТОВ.

На следующем примере (рис. 5.3) изображена рекурсивная связь, связывающая сущность МУЖЧИНА с ней же самой. Конец связи с именем «сын» определяет тот факт, что несколько людей могут быть сыновьями одного отца. Конец связи с именем «отец» означает, что не у каждого мужчины должны быть сыновья.

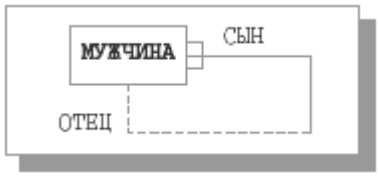

Рис. 5.3. Пример рекурсивного типа связи

Лаконичная устная трактовка изображенной диаграммы состоит в следующем:

- каждый МУЖЧИНА является сыном одного и только одного МУЖЧИНЫ;
- каждый МУЖЧИНА может являться отцом одного или более МУЖЧИН.

Атрибутом сущности является любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности. Имена атрибутов заносятся в прямоугольник, изображающий сущность, под именем сущности и изображаются малыми буквами, возможно, с примерами.

Пример типа сущности ЧЕЛОВЕК с указанными атрибутами показан на рис. 5.4. С технической точки зрения атрибуты типа сущности в ER-модели похожи на атрибуты отношения в реляционной модели данных. И в том, и в другом случаях введение именованных атрибутов вводит некоторую типовую структуру данных, имя которой совпадает с именем типа сущности в случае ER-модели или с именем переменной отношения в случае реляционной модели. Этой типовой структуре должны следовать все экземпляры типа сущности или все кортежи отношения. Но имеется и важное отличие. Напомним, что в реляционной модели данных атрибут определяется как упорядоченная пара <имя_атрибута, имя_домена> (или <имя_атрибута, имя_базового_типа_данных>, если понятие домена не поддерживается). Заголовок отношения, определяемый как множество таких пар, представляет собой полный аналог структурного типа данных в языках программирования.

Рис. 5.4. Пример типа сущности с атрибутами

При определении атрибутов типа сущности в ER-модели указание домена атрибута не является обязательным, хотя это и возможно (см. ниже). Обсудим, чем вызвана эта возможность «ослабленного» определения атрибутов. Прежде всего, как уже отмечалось, семантические модели данных используются для построения концептуальных схем БД, и эти схемы преобразуются в реляционные схемы БД, которые поддерживаются той или иной СУБД. Несмотря на то, что в настоящее время типовые возможности РСУБД в основном стандартизованы (на основе стандарта языка SQL), детали базового набора типов данных и средств определения доменов в разных системах могут различаться. Поскольку производители CASE-средств проектирования реляционных БД стремятся не связывать обеспечиваемые ими возможности семантического моделирования с конкретной реализацией СУБД, они стимулируют откладывание строгого определения типов атрибутов до стадии полного определения реляционной схемы.

Кроме того, напомним, что при определении атрибута отношения допускается использование имен атрибутов, совпадающих с именами своих доменов (это два разных пространства имен, и наличие одинаковых имен у атрибутов и доменов не вызывает коллизий). Поэтому при определении атрибутов типов сущности можно так подбирать их имена, что они в дальнейшем будут подсказывать, какие домены у этих атрибутов имеются в виду. Пониманию предполагаемой сути доменов способствует и возможность указания примеров значений атрибутов. Например, на рис. 5.4 имеется атрибут год рождения, в качестве примерного значения которого указано «1976». Это подсказывает, что в реляционной схеме при определении соответствующего атрибута наиболее естественным базовым типом данных будет темпоральный тип «ДАТА», значения которого задают дату с точностью до года.

Уникальные идентификаторы типов сущности

Как отмечалось выше, при определении типа сущности необходимо гарантировать, что каждый экземпляр сущности является отличимым от любого другого экземпляра той же сущности. Поскольку сущность является абстракцией реального или представляемого объекта внешнего мира, это требование нужно иметь в виду уже при выборе кандидата в типы сущности. Например, предположим, что проектируется база данных для поддержки работы книжного склада. На складе могут храниться произвольные части тиража любого издания любой книги. Может ли в этом случае индивидуальная книга являться прообразом типа сущности? Утверждается, что нет, поскольку отсутствует возможность различения книг одного издания. Для книжного склада прообразом типа сущности будет набор одноименных книг одного автора, вышедших в одном издании. Одним из атрибутов этого типа сущности будет число книг в наборе. Но когда книга поступает в библиотеку и ей присваивается уникальный библиотечный номер, она становится разумным прообразом типа сущности. Плохо устроены библиотеки, в которых не различаются индивидуальные книги (даже одноименные книги одного автора, вышедшие в одном издании).

Но при проектировании базы данных мало того, чтобы проектировщик убедился в правильном выборе типов сущности, гарантирующем различие экземпляров каждого типа сущности. Необходимо сообщить системе автоматизации проектирования БД, каким образом будут различаться эти экземпляры, т. е. сообщить, как конструируются уникальные идентификаторы экземпляров каждого типа сущности. В ER-модели у экземпляра типа сущности не может быть назначаемого пользователем имени или назначаемого системой внешнего уникального идентификатора. Экземпляр типа сущности может идентифицироваться только своими индивидуальными характеристиками, а они представляются значениями атрибутов и экземплярами типов связи, связывающими данный экземпляр типа сущности с экземплярами других типов сущности или этого же типа сущности. Поэтому уникальным идентификатором сущности может быть атрибут, комбинация атрибутов, связь, комбинация связей или комбинация связей и атрибутов, уникально отличающая любой экземпляр сущности от других экземпляров сущности того же типа.

Приведем несколько примеров. На рис. 5.5 показан тип сущности КНИГА, пригодный для использования в базе данных книжного склада. При издании любой книги в любом издательстве (кроме пиратских, которыми мы для простоты пренебрежем) ей присваивается уникальный номер – ISBN. Понятно, что значение атрибута isbn будет уникально идентифицировать партию книг на складе. Кроме того, конечно, в качестве уникального идентификатора годится и комбинация атрибутов <автор, название, номер издания, издательство, год издания>.

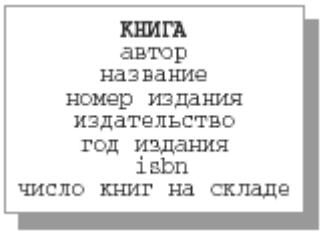

Рис. 5.5. Тип сущности, экземпляры которого идентифицируются атрибутами

На рис. 10.6 диаграмма включает два связанных типа сущности. У каждого взрослого человека имеется один и только один паспорт (мы снова не берем в расчет особый случай, когда у одного человека имеется несколько паспортов), и каждый паспорт может принадлежать только одному взрослому человеку (некоторые уже готовые паспорта могут быть еще никому не выданы). Тогда связь человека с его паспортом (конец связи ИМЕЕТ) уникально идентифицирует взрослого человека, т. е., грубо говоря, паспорт определяет взрослого человека. Поскольку могут существовать паспорта, еще не выданные какому-либо человеку, эта связь не является уникальным идентификатором сущности ПАСПОРТ.

Рис. 5.6. Тип сущности, экземпляры которого идентифицируются связью

На рис. 5.7 диаграмма включает три связанных типа сущности. Профессора обладают знаниями в нескольких учебных дисциплинах. Преподавание каждой дисциплины доступно нескольким профессорам. Другими словами, между сущностями ПРОФЕССОР и ДИСЦИПЛИНА определена связь «многие ко многим». Каждый профессор может готовить курсы по любой доступной ему дисциплине. Каждой дисциплине может быть посвящено несколько учебных курсов. Но каждый профессор может готовить только один курс по любой доступной ему дисциплине, и каждый курс может быть посвящен только одной дисциплине. Тем самым, каждый экземпляр типа сущности КУРС уникально идентифицируется экземпляром сущности ПРОФЕССОР и экземпляром сущности ДИСЦИПЛИНА, т. е. парой связей с именами концов ГОТОВИТСЯ и ПОСВЯЩЕН на стороне сущности КУРС. Заметим, что сущности ПРОФЕССОР и ДИСЦИПЛИНА связями не идентифицируются.

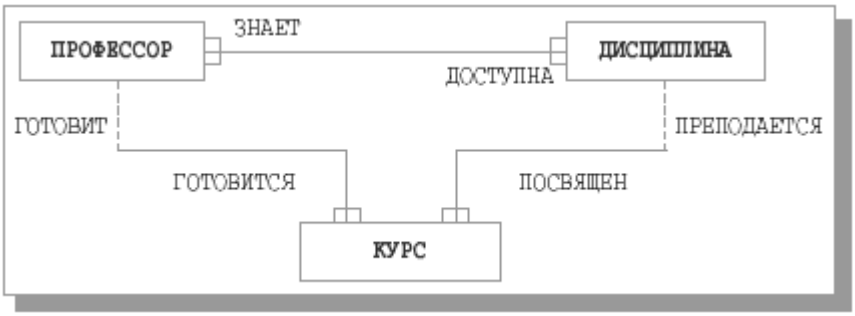

Рис. 5.7. Тип сущности, экземпляры которого идентифицируются комбинацией связей

Наконец, на рис. 5.8 приведен пример типа сущности, уникальный идентификатор которого является комбинацией атрибутов и связей. Это несколько уточненный вариант сущности с рекурсивной связью с рис. 5.3. У каждого человека могут быть дети, и у каждого человека имеется отец. Тогда, если предположить, что близнецам, появившимся на свет одновременно, не дают одинаковых имен, то уникальным идентификатором типа сущности ЧЕЛОВЕК может быть комбинация атрибутов <дата рождения, ФИО> и связь с именем конца РЕБЕНОК.

Рис.5.8. Тип сущности, экземпляры которого идентифицируются комбинацией атрибутов и связей

Понятно, что это «определение» на самом деле является тавтологией, поскольку, во-первых, мы пытаемся определить термин сущность через не определенный термин объект, а во-вторых, попытки определения термина объект настолько же безнадежны. Обычно авторы пытаются оправдываться тем, что в подобном контексте они имеют в виду «житейское», а не сколько-нибудь формализованное понятие объекта.

Конечно, от этого не становится легче, поскольку понятие сущности должно пониматься в достаточно точном смысле. Но эта тавтология не изобретена автором этого курса; она традиционна для области семантического моделирования. В этой области стремятся максимально избегать формальностей.

Хотя было бы правильнее всегда использовать термины тип сущности и экземпляр типа сущности, для избежания многословности (и следуя традиции) в тех случаях, где это не приводит к двусмысленности, мы будем использовать термин сущность в значении типа сущности.

Тем не менее, как и в случае типа сущности, мы будем часто использовать термин связь в значении типа связи.

В некоторых вариантах ER-модели конец связи называют ролью связи в данной сущности. Тогда можно говорить об имени роли, степени роли и обязательности роли связи в данной сущности.

Нормальные формы ER-диаграмм

Как и в случае схем реляционных баз данных, для ER-диаграмм вводится понятие нормальных форм, причем их смысл очень близко соответствует смыслу нормальных форм отношений. Заметим, что определения нормальных форм ER-диаграмм делают более понятным смысл нормализации схем отношений. Мы приведем только очень краткие и неформальные определения трех первых нормальных форм. Конечно, можно было бы ввести дальнейшие нормальные формы ER-диаграмм, аналогичные нормальной форме Бойса-Кодда, 4NF и 5NF, но на практике к такой нормализации обычно не прибегают, а общие идеи после ознакомления с лекцией 9 должны быть понятны и так.

Первая нормальная форма ER-диаграммы

В первой нормальной форме ER-диаграммы устраняются атрибуты, содержащие множественные значения, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

На рис. 5.9 (а) показана диаграмма, в которой тип сущности АЭРОДРОМ не удовлетворяет требованию первой нормальной формы. Здесь для нас несущественны атрибуты сущности АВИАРЕМОНТНОЕ ПРЕДПРИЯТИЕ, но сущность АЭРОДРОМ помимо атрибутов, отражающих собственные характеристики аэродромов (длина взлетно-посадочной полосы, число ангаров и т.д.) содержит атрибут, множественное значение которого характеризует самолеты, приписанные к этому аэродрому. Очевидно, что самолеты нуждаются в ремонте, т. е. должны обслуживаться некоторым авиаремонтным предприятием. Но поскольку самолеты являются частью сущности АЭРОДРОМ, единственным способом фиксации этого факта на диаграмме является проведение связи «многие ко многим» между типами сущности АЭРОДРОМ и АВИАРЕМОНТНОЕ ПРЕДПРИЯТИЕ. Таким образом выражается то соображение, что для ремонта разных самолетов, приписанных к одному аэродрому, могут использоваться разные транспортные предприятия, и каждое транспортное предприятие может обслуживать несколько аэродромов.

Рис. 5.9. Пример приведения ER-диаграммы к первой нормальной форме

Чем плоха эта ситуация? Прежде всего, тем, что скрывается тот факт, что авиаремонтное предприятие ремонтирует самолеты, а не аэродромы. Наша же связь на самом деле означает, что любой аэродром из группы аэродромов обслуживается любым авиаремонтным предприятием из группы таких предприятий. Проблема состоит именно в том, что значением атрибута «самолеты» является множество экземпляров типа сущности САМОЛЕТ, и этот тип сущности сам обладает атрибутами и связями.

Ситуацию исправляет ER-диаграмма, показанная на рис. 5.9 (b). Здесь мы выделили тип сущности САМОЛЕТ. Связь между сущностями АЭРОПОРТ и САМОЛЕТ показывает, что к одному аэродрому приписывается несколько самолетов. Связь между сущностями САМОЛЕТ и АВИАРЕМОНТНОЕ ПРЕДПРИЯТИЕ означает, что каждый самолет из группы самолетов (группу самолетов могут составлять, например, все самолеты одного типа) обслуживается любым транспортным предприятием из некоторой группы таких предприятий. ER-диаграмма на рис. 10.9 (b) находится в первой нормальной форме и, как мы видим, лучше отображает реальную ситуацию.

Вторая нормальная форма ER-диаграммы

Во второй нормальной форме устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

На рис. 5.10 (a) показана диаграмма, на которой тип сущности ЭЛЕМЕНТ РАСПИСАНИЯ не удовлетворяет требованиям второй нормальной формы. На этой диаграмме у сущности ЭЛЕМЕНТ РАСПИСАНИЯ имеются следующие свойства. Элементы расписания предназначены для сохранения данных о рейсах самолетов, вылетающих в течение дня. Некоторыми важными характеристиками рейса являются номер рейса, аэропорт вылета, аэропорт назначения, дата и время вылета, бортовой номер самолета, тип самолета. Если говорить про российские авиационные компании, то (1) у каждого рейса имеется заранее приписанный ему номер (уникальный среди всех других имеющихся номеров рейсов), (2) не все рейсы совершаются каждый день, поэтому характеристикой конкретного рейса является дата и время его совершения, (3) бортовой номер самолета определяется парой <номер рейса, дата-время вылета>. Имеется связь «многие к одному» между сущностями ЭЛЕМЕНТ РАСПИСАНИЯ и ГОРОД. Экземпляры типа сущности ГОРОД характеризуют город, в который прибывает данный рейс.

Рис. 5.10. Пример приведения ER-диаграммы ко второй нормальной форме

Уникальным идентификатором типа сущности ЭЛЕМЕНТ РАСПИСАНИЯ является пара атрибутов <номер рейса, дата-время вылета>. Если вернуться к терминам функциональных зависимостей, то между атрибутами этой сущности имеются следующие FD:

- {номер рейса, дата-время вылета} ⟶ бортовой номер самолета;
- номер рейса ⟶ аэропорт вылета;
- номер рейса ⟶ аэропорт назначения;
- бортовой номер самолета ⟶ тип самолета.

Кроме того, очевидно, что каждый экземпляр связи с сущностью ГОРОД также определяется значением атрибута номер рейса. Налицо нарушение требования второй нормальной формы. Мы получаем не только избыточное хранение значений атрибутов аэропорт вылета и аэропорт назначения в каждом экземпляре типа сущности ЭЛЕМЕНТ РАСПИСАНИЯ с одним и тем же значением номера рейса. Искажается и затемняется смысл связи с сущностью ГОРОД. Можно подумать, что в разные дни один и тот же рейс прибывает в разные города.

На рис. 10.10 (b) показан нормализованный вариант диаграммы, в котором все сущности находятся во второй нормальной форме. Теперь имеются три типа сущности: РЕЙС с атрибутами номер

рейса, аэропорт вылета, аэропорт назначения, ЭЛЕМЕНТ РАСПИСАНИЯ с атрибутами дата-время вылета, бортовой номер самолета, тип самолета и ГОРОД. Уникальным идентификатором сущности РЕЙС является атрибут номер рейса, уникальный идентификатор ЭЛЕМЕНТ РАСПИСАНИЯ состоит из атрибута дата вылета и конца связи КОГДА, НА ЧЕМ. Мы видим, что ни в одном типе сущности больше нет атрибутов, определяемых частью уникального идентификатора. Свойства второй нормальной формы удовлетворяются, и мы имеем более качественную диаграмму.

Третья нормальная форма ER-диаграммы

- В третьей нормальной форме устраняются атрибуты, которые зависят от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

Взглянем еще раз на тип сущности ЭЛЕМЕНТ РАСПИСАНИЯ на рис. 5.10 (b). Конечно, каждый день каждый рейс выполняется только одним самолетом, поэтому бортовой номер самолета полностью зависит от уникального идентификатора. Но бортовой номер является уникальной характеристикой каждого самолета, и от этой характеристики зависят все остальные характеристики, в частности, тип самолета. Другими словами, между уникальным идентификатором и другими атрибутами типа сущности ЭЛЕМЕНТ РАСПИСАНИЯ имеются следующие функциональные зависимости:

{КОГДА, НА ЧЕМ, дата-время вылета} ⟶ бортовой номер самолета

{КОГДА, НА ЧЕМ, дата-время вылета} ⟶ тип самолета

бортовой номер самолета ⟶ тип самолета

Как видно, имеется транзитивная FD {КОГДА, НА ЧЕМ, дата вылета} ⟶ тип самолета, и наличие этой FD вызывает нарушение требования третьей нормальной формы. На самом деле, тип сущности ЭЛЕМЕНТ РАСПИСАНИЯ на рис. 5.10 (b) включает в себя (по крайней мере, частично) тип сущности САМОЛЕТ. Это вызывает избыточность хранения и затуманивает смысл диаграммы. На рис. 5.11 показан нормализованный вариант диаграммы, в котором все сущности находятся в третьей нормальной форме.

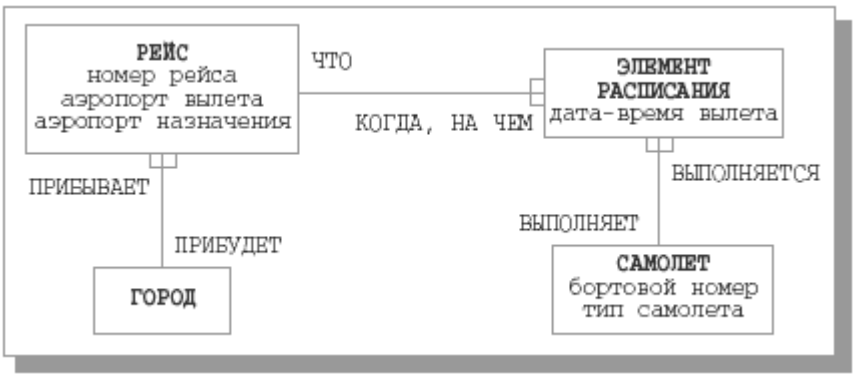

Рис. 5.11. Пример приведения ER-диаграммы к третьей нормальной форме

Более сложные элементы ER-модели

До сих пор мы рассматривали только самые основные и наиболее очевидные понятия ER-модели данных. К числу некоторых более сложных элементов модели относятся следующие.

- Подтипы и супертипы сущностей. Подобно тому как это делается в языках программирования с развитыми типовыми системами (например, в языках объектно-ориентированного программирования), в ER-модели поддерживается возможность определения нового типа сущности путем наследования некоторого супертипа сущности. Механизм наследования в ER-модели обладает несколькими особенностями: в частности, интересные нюансы связаны с необходимостью графического изображения этого механизма (см. ниже).
- Уточняемые степени связи. Иногда бывает полезно определить возможное количество экземпляров сущности, участвующих в данной связи (например, ввести ограничение, связанное с тем, что служащему разрешается участвовать не более чем в трех проектах одновременно). Для выражения этого семантического ограничения разрешается указывать на конце связи ее максимально допустимую или обязательную степень.
- Взаимно исключающие связи. Для заданного типа сущности можно определить такой набор типов связи с другими типами сущности, что для каждого экземпляра заданного типа сущности может (если набор связей является необязательным) или должен (если набор связей обязателен) существовать экземпляр только одной связи из этого набора.
- Каскадные удаления экземпляров сущностей. Некоторые связи бывают настолько сильными (конечно, в случае связи «один ко многим»), что при удалении опорного экземпляра сущности (соответствующего концу связи «один») нужно удалить и все экземпляры сущности, соответствующие

концу связи «многие». Соответствующее требование каскадного удаления можно специфицировать при определении связи.

- Домены. Как и в случае реляционной модели данных, в некоторых случаях полезна возможность определения потенциально допустимого множества значений атрибута сущности (домена).

Эти и другие усложненные элементы модели данных «Сущность-Связь» делают ее более мощной, но одновременно несколько затрудняют ее использование. Конечно, при реальном применении ER-диаграмм для проектирования баз данных необходимо ознакомиться со всеми возможностями. Ниже мы подробнее обсудим два элемента из числа упомянутых выше – супертипы и подтипы сущности, а также приведем пример сущности с взаимно исключающими связями.

Наследование типов сущности и типов связи

Сущность может быть расщеплена на два или большее число взаимно исключающих подтипов, каждый из которых включает общие атрибуты и/или связи. Эти общие атрибуты и/или связи явно определяются один раз на более высоком уровне. В подтипах могут определяться собственные атрибуты и/или связи. В принципе, подтипизация может продолжаться на более низких уровнях, но опыт использования ER-модели при проектировании баз данных показывает, что в большинстве случаев оказывается достаточно двух-трех уровней.

Если у типа сущности A имеются подтипы B1, B2,..., Bn, то:

- любой экземпляр типа сущности B1, B2,..., Bn является экземпляром типа сущности A (включение);
- если a является экземпляром типа сущности A, то a является экземпляром некоторого подтипа сущности Bi (i = 1, 2, ..., n) (отсутствие собственных экземпляров у супертипа сущности);
- ни для каких подтипов Bi и Bj (i, j = 1, 2, ..., n) не существует экземпляра, типом которого одновременно являются типы сущности Bi и Bj (разъединенность подтипов).

Тип сущности, на основе которого определяются подтипы, называется супертипом. Как мы видели выше, подтипы должны образовывать полное множество, т. е. любой экземпляр супертипа должен относиться к некоторому подтипу. Иногда для обеспечения такой полноты приходится определять дополнительный подтип ПРОЧИЕ.

Пример супертипа ЛЕТАТЕЛЬНЫЙ АППАРАТ и его подтипов АЭРОПЛАН, ВЕРТОЛЕТ, ПТИЦЕЛЕТ и ПРОЧИЕ показан на рис. 5.12. У подтипа АЭРОПЛАН имеются два собственных подтипа – ПЛАНЕР и МОТОРНЫЙ САМОЛЕТ. Для супертипа сущности ЛЕТАТЕЛЬНЫЙ АППАРАТ определен атрибут максимальная дальность полета и необязательная связь «многие ко многим» с типом сущности ПИЛОТ. Эти атрибут и связь наследуется всеми подтипами этого супертипа сущности. У непосредственного подтипа сущности АЭРОПЛАН определяется один дополнительный атрибут, так что в совокупности у данного типа сущности имеются два атрибута максимальная дальность полета и размах крыльев и одна унаследованная связь с типом сущности ПИЛОТ. У подтипа второго уровня МОТОРНЫЙ САМОЛЕТ супертипа АЭРОПЛАН определяется один дополнительный атрибут мощность мотора и одна дополнительная (обязательная) связь с типом сущности АЭРОДРОМ. Тем самым, у типа сущности МОТОРНЫЙ САМОЛЕТ имеются три атрибута: два унаследованных – максимальная дальность полета и размах крыльев и один собственный – мощность мотора, а также две связи: одна унаследованная – с типом сущности ПИЛОТ и одна собственная – с типом сущности АЭРОДРОМ. И так далее. Понятно, что для типа сущности ПРОЧИЕ, скорее всего, бессмысленно определять собственные атрибуты и связи, так что свойства этого типа будут совпадать со свойствами его супертипа.

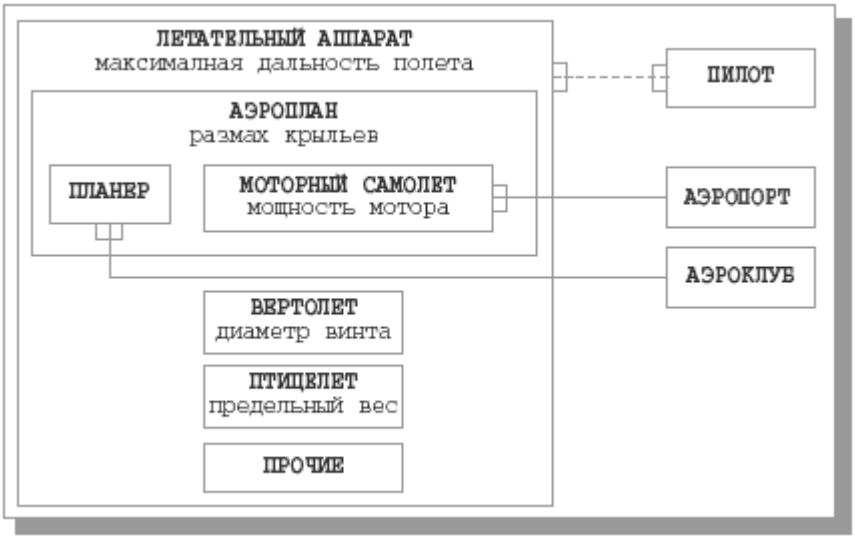

Рис. 5.12. Супертипы и подтипы сущности

Как же следует понимать диаграмму, представленную на рис. 10.12? Если начинать от супертипа, то диаграмма изображает ЛЕТАТЕЛЬНЫЙ АППАРАТ, который должен быть АЭРОПЛАНОМ, ВЕРТОЛЕТОМ, ПТИЦЕЛЕТОМ или ДРУГИМ ЛЕТАТЕЛЬНЫМ АППАРАТОМ. Если начинать от подтипа (например, сущности ВЕРТОЛЕТ), то это ВЕРТОЛЕТ, который относится к типу ЛЕТАТЕЛЬНОГО АППАРАТА. Если начинать от подтипа, который является одновременно супертипом, то это АЭРОПЛАН, который относится к типу ЛЕТАТЕЛЬНОГО АППАРАТА и должен быть ПЛАНЕРОМ или МОТОРНЫМ САМОЛЕТОМ.

В механизме наследования ER-модели допускается наличие двух или более разбиений сущности на подтипы. Например, тип сущности ЧЕЛОВЕК может быть расщеплен на подтипы по профессиональному признаку (ПРОГРАММИСТ, ДОЯРКА и т. д.), а может быть расщеплен и по половому признаку (МУЖЧИНА, ЖЕНЩИНА).

Взаимно исключающие связи

Пример диаграммы из двух сущностей с взаимно исключающими связями показан на рис. 10.13(а). Самолет может находиться в рабочем состоянии, и тогда у него имеется один и только один пилот. Или же самолет может находиться на ремонте на одном из нескольких возможных авиаремонтных предприятий (каждое предприятие может производить ремонт нескольких самолетов).

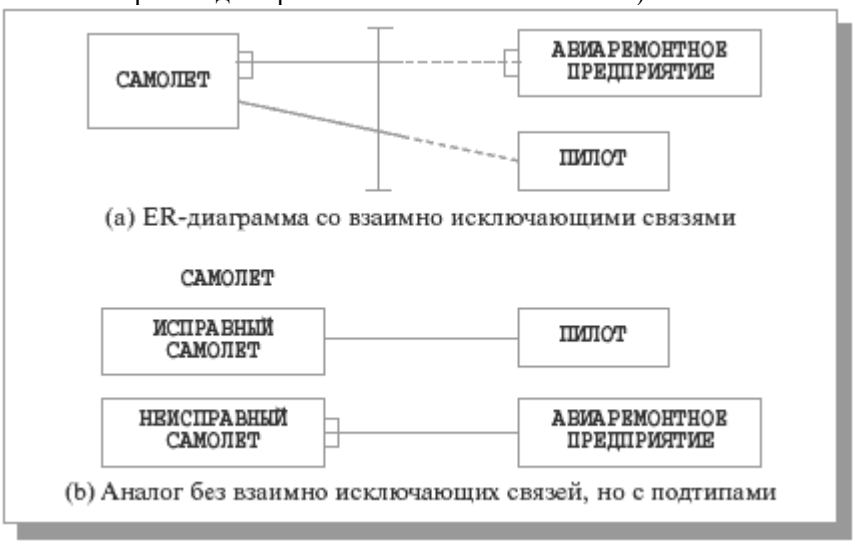

Рис. 5.13. Пример ER-диаграммы со взаимно исключающими связями

В данном случае для каждого экземпляра типа сущности САМОЛЕТ должен существовать экземпляр одной из указанных связей. Для экземпляров типа сущности САМОЛЕТ, соответствующих исправным самолетам, должен существовать экземпляр связи «один к одному» с экземпляром типа сущности ПИЛОТ, а экземпляры, соответствующие неисправным самолетам, должны участвовать в экземпляре типа связи «многие к одному» с экземпляром типа сущности АВИАРЕМОНТНОЕ ПРЕДПРИЯТИЕ.

Как показано на рис. 5.13(b), диаграмма со взаимно исключающими связями из рис. 5.13(а) может быть преобразована к диаграмме без взаимно исключающих связей путем введения подтипов. Поскольку

любой самолет может быть либо исправным, либо неисправным, можно корректным образом ввести два подтипа супертипа САМОЛЕТ – ИСПРАВНЫЙ САМОЛЕТ и НЕИСПРАВНЫЙ САМОЛЕТ. На уровне супертипа сущности связи не определяются. Для подтипа ИСПРАВНЫЙ САМОЛЕТ определяется обязательная связь «один к одному» с типом сущности ПИЛОТ, а для подтипа НЕИСПРАВНЫЙ САМОЛЕТ определяется обязательная связь «многие к одному» с типом сущности АВИАРЕМОНТНОЕ ПРЕДПРИЯТИЕ.

Заметим, что для того чтобы описанная схема реализации механизма взаимно исключающих связей на основе механизма наследования действительно могла работать, в средствах манипулирования данными ER-модели должна быть предусмотрена возможность динамического изменения типа сущности у экземпляра. Конкретно для нашего случая требуется возможность изменения типа экземпляра сущности ИСПРАВНЫЙ САМОЛЕТ на тип сущности НЕИСПРАВНЫЙ САМОЛЕТ, и наоборот (исправный самолет может ломаться, неисправный самолет – приводиться в рабочее состояние). Конечно, при такой смене типа должен изменяться и экземпляр связи. Заметим, что в рассматриваемом случае мы имеем дело с ограниченным динамическим изменением типа экземпляра, поскольку и исправные, и неисправные самолеты являются экземплярами супертипа САМОЛЕТ.

Получение реляционной схемы из ER-диаграммы

Опишем типовую многошаговую процедуру преобразования ER-диаграммы в реляционную (более точно, в SQL-ориентированную) схему базы данных.

Базовые приемы

Каждый простой тип сущности превращается в таблицу. (Простым типом сущности называется тип сущности, не являющийся подтипом и не имеющий подтипов.) Имя сущности становится именем таблицы. Экземплярам типа сущности соответствуют строки соответствующей таблицы.

Каждый атрибут становится столбцом таблицы с тем же именем; может выбираться более точный формат представления данных. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, – не могут.

Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификаторов, для первичного ключа выбирается наиболее характерный. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно, и в общем случае может привести к зацикливанию). Для именования этих столбцов используются имена концов связей и/или имена парных типов сущностей.

Связи «многие к одному» (и «один к одному») становятся внешними ключами, т. е. образуется копия уникального идентификатора сущности на конце связи «один», и соответствующие столбцы составляют внешний ключ таблицы, соответствующей типу сущности на конце связи «многие». Необязательные связи соответствуют столбцам внешнего ключа, допускающим наличие неопределенных значений; обязательные связи – столбцам, не допускающим неопределенных значений. Если между двумя типами сущности A и B имеется связь «один к одному», то соответствующий внешний ключ по желанию проектировщика может быть объявлен как в таблице A, так и в таблице B. Чтобы отразить в определении таблицы ограничение, которое заключается в том, что степень конца связи должна равняться единице, соответствующий (возможно, составной) столбец должен быть дополнительно специфицирован как возможный ключ таблицы (в случае использования языка SQL для этого служит спецификация UNIQUE – см. лекцию 16).

Для поддержки связи «многие ко многим» между типами сущности A и B создается дополнительная таблица AB с двумя столбцами, один из которых содержит уникальные идентификаторы экземпляров сущности A, а другой – уникальные идентификаторы экземпляров сущности B. Обозначим через УИД(c) уникальный идентификатор экземпляра с некоторого типа сущности C. Тогда, если в экземпляре связи «многие ко многим» участвуют экземпляры a1, a2, ..., an типа сущности A и экземпляры b1, b2, ..., bm типа сущности B, то в таблице AB должны присутствовать все строки вида {УИД(ai), УИД(bj)} для i = 1, 2, ..., nn, j = 1, 2, ..., m. Понятно, что, используя таблицы A, B и AB, с помощью стандартных реляционных операций можно найти все пары экземпляров типов сущности, участвующих в данной связи.

Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы.

Представление в реляционной схеме супертипов и подтипов сущности

В этом подразделе мы предполагаем, что реляционная схема базы данных проектируется в расчете на использование обычной SQL-ориентированной СУБД, не поддерживающей объектно-реляционные расширения. Кстати, заметим, что поддержка таких расширений не слишком помогает при переходе от

концептуальной схемы базы данных в модели «Сущность-Связь» к объектно-реляционной схеме, соответствующей последним стандартам языка SQL.

Если в концептуальной схеме (ER-диаграмме) присутствуют подтипы, то возможны два способа их представления в реляционной схеме:

(a) собрать все подтипы в одной таблице;

(b) для каждого подтипа образовать отдельную таблицу.

При применении способа (a) таблица создается для максимального супертипа (типа сущности, не являющегося подтипом), а для подтипов могут создаваться представления. Таблица содержит столбцы, соответствующие каждому атрибуту (и связям) каждого подтипа. В таблицу добавляется, по крайней мере, один столбец, содержащий код ТИПА; он становится частью первичного ключа. Для каждой строки таблицы значение этого столбца определяет тип сущности, экземпляру которого соответствует строка. Столбцы этой строки, которые соответствуют атрибутам и связям, отсутствующим в данном типе сущности, должны содержать неопределенные значения.

При использовании метода (b) для каждого подтипа первого уровня (для более глубоких уровней применяется метод (a)) супертип воссоздается с помощью представления UNION (из всех таблиц подтипов выбираются общие столбцы – столбцы супертипа).

- У каждого способа есть свои достоинства и недостатки. К достоинствам первого способа (одна таблица для супертипа и всех его подтипов) можно отнести следующее:
- соответствие логике супертипов и подтипов; поскольку любой экземпляр любого подтипа является экземпляром супертипа, логично хранить вместе все строки, соответствующие экземплярам супертипа;
- обеспечение простого доступа к экземплярам супертипа и не слишком сложный доступ к экземплярам подтипов;
- возможность обойтись небольшим числом таблиц.

Недостатки метода (a):

- прикладная программа, имеющая дело с одной таблицей супертипа, должна включать дополнительную логику работы с разными наборами столбцов (в зависимости от значения столбца ТИП) и разными ограничениями целостности (в зависимости от особенностей связей, определенных для подтипа);
- общая для всех подтипов таблица потенциально может стать узким местом при многопользовательском доступе по причине возможности блокировки таблицы целиком;
- для индивидуальных столбцов подтипов должна допускаться возможность содержать неопределенные значения; таким образом, потенциально в общей таблице будет содержаться много неопределенных значений, что при использовании некоторых РСУБД может потребовать значительного объема внешней памяти.

Достоинства метода (b) состоят в следующем:

- действуют более понятные правила работы с подтипами (каждому подтипу соответствует одноименная таблица);
- упрощается логика приложений; каждая программа работает только с нужной таблицей.

Недостатки метода (b):

- в общем случае требуется слишком много отдельных таблиц;
- работа с экземплярами супертипа на основе представления, объединяющего таблицы супертипов, может оказаться недостаточно эффективной;
- поскольку множество экземпляров супертипа является объединением множеств экземпляров подтипов, не все РСУБД могут обеспечить выполнение операций модификации экземпляров супертипа.

Представление в реляционной схеме взаимно исключающих связей

Существуют два способа формирования схемы реляционной БД при наличии взаимно исключающих связей (имеются в виду связи «один ко многим», причем конец связи «многие» находится на стороне сущности, для которой связи являются взаимно исключающими):

(a) общее хранение внешних ключей;

(b) раздельное хранение внешних ключей.

Понятно, что если имеются взаимно исключающие связи упомянутой категории, то в таблице, соответствующей сущности, для которой связи являются взаимно исключающими, необходимо хранить внешние ключи. Если внешние ключи всех потенциально связанных таблиц имеют общий формат, то можно применить способ (a), т. е. создать два столбца: идентификатор связи и идентификатор сущности (возможно, составной). Столбец идентификатора связи используется для различения связей, покрываемых дугой исключения. В столбце (столбцах) идентификатора сущности хранятся значения уникального идентификатора сущности на дальнем конце соответствующей связи.

Если результирующие внешние ключи не относятся к одному домену, то приходится прибегать к использованию способа (b), т. е. создавать для каждой связи, покрываемой дугой исключения, явные столбцы внешних ключей; все эти столбцы могут содержать неопределенные значения.

Преимущество подхода (a) состоит в том, что в таблице, соответствующей сущности, появляется всего два дополнительных столбца. Очевидным недостатком является усложнение выполнения операции соединения: чтобы воспользоваться для соединения одной из альтернативных связей, нужно сначала произвести ограничение таблицы в соответствии с нужным значением столбца, содержащего идентификаторы связей.

При использовании подхода (b) соединения являются явными (и естественными). Недостаток состоит в том, что требуется иметь столько столбцов, сколько имеется альтернативных связей. Кроме того, в каждом из таких столбцов будет содержаться много неопределенных значений, хранение которых потенциально может привести к серьезным накладным расходам внешней памяти.

Рис. 5.14. Возможные модификации ER-диаграмм, позволяющие избежать взаимно исключающих связей

Модификация, показанная на рис. 5.14 (b), основана на том наблюдении, что коль скоро связи являются альтернативными, то они разделяют множество экземпляров сущности A на два или более непересекающихся подмножества, которые могут лежать в основе определения подтипов A1 и A2. Это хороший вариант, если такие подтипы могут пригодиться еще для чего-нибудь. Например, в случае взаимно исключающей связи, представленной на рис. 5.12, у исправных и неисправных самолетов могут имется несовпадающие множества атрибутов (скажем, у типа сущности ИСПРАВНЫЕ САМОЛЕТЫ может иметься атрибут дата завершения гарантийного срока, а у типа сущности НЕИСПРАВНЫЕ САМОЛЕТЫ – атрибут тип неисправности). С другой стороны, как отмечалось в предыдущем разделе, для использования этого подхода требуется возможность динамического изменения типа существующего экземпляра.

Модификация, показанная на рис. 5.14 (c), основана на том наблюдении, что коль скоро типы сущности B и C участвуют в альтернативной связи, то, по всей видимости, у этих сущностей имеется что-то общее. Возможно, их было бы правильнее определять как подтипы некоторого общего типа сущности. Заметим, что пример с рис. 5.12 явно демонстрирует, что далеко не всегда типы сущности, участвующие в альтернативной связи, обладают общими чертами. Создание общего супертипа для типов сущности ПИЛОТ и АВИАРЕМОНТНОЕ ПРЕДПРИЯТИЕ представляется весьма странной идеей.

На этом мы заканчиваем краткую экскурсию в семантическое моделирование с использованием ER-диаграмм.

Заключение

Основной целью данной лекции было ознакомление с семантическими моделями данных на примере упрощенного варианта ER-модели. Представленный вариант ER-модели, с одной стороны, является достаточно развитым, чтобы можно было почувствовать общую специфику семантических моделей данных, а с другой стороны, не перегружен деталями и излишними понятиями, затрудняющими общее понимание подхода.

С практической точки зрения наибольшую пользу могут принести рассмотренные приемы перехода от ER-диаграмм к схеме реляционной базы данных. Особенно могут пригодиться рекомендации по

представлению в реляционной схеме связей «многие ко многим», подтипов и супертипов сущности и взаимно исключающих связей.

Тема 7. Язык SQL

SQL: история языка

Одним из наиболее важных шагов на пути к признанию SQL на рынке стало появление стандартов на этот язык. Обычно при упоминании стандарта SQL имеют в виду официальный стандарт, утвержденный Американским институтом национальных стандартов (American National Standards Institute – ANSI) и Международной организацией по стандартам (International Standards Organization – ISO). Однако существуют и другие важные стандарты SQL, включая SQL, реализованный в системе DB2 компании IBM, и стандарт X/OPEN для SQL в среде UNIX.

Стандарт SQL ANSI/ISO

Работа над официальным стандартом SQL началась в 1982 году, когда ANSI поставил перед своим комитетом ХЗН2 задачу по созданию стандарта языка реляционных баз данных. Вначале в комитете обсуждались достоинства различных предложенных языков. Однако поскольку к тому времени SQL уже стал фактическим стандартом, комитет ХЗН2 остановил свой выбор на нем и занялся стандартизацией SQL.

Разработанный в результате стандарт в большой степени был основан на диалекте SQL системы DB/2, хотя и содержал в себе ряд существенных отличий от этого диалекта. После нескольких доработок, в 1986 году стандарт был официально утвержден как стандарт ANSI номер Х3.135, а в 1987 году — в качестве стандарта ISO. Затем стандарт ANSI/ISO был принят правительством США как федеральный стандарт США по обработке информации (FIPS — Federal Information Processing Standard). Этот стандарт, незначительно пересмотренный в 1989 году, обычно называют стандартом "SQL-89", ил"SQLI". Когда в данной книге упоминается "стандарт ANSI/ISO", то подразумевается SQLI, который в настоящее время лежит в основе большинства коммерческих продуктов.

Многие из членов комитетов по стандартизации ANSI и ISO представляли фирмы- поставщики различных СУБД, в каждой из которых был реализован собственный диалект SQL. Как и диалекты человеческого языка, диалекты SQL были в основном похожи друг на друга, однако несовместимы в деталях. Во многих случаях комитет просто обошел существующие различия и не стандартизировал некоторые части языка, определив, что они реализуются по усмотрению разработчика. Этот подход позволил объявить большое число реализаций SQL совместимыми со стандартом, однако сделал сам стандарт относительно слабым.

Чтобы заполнить эти пробелы, комитет ANSI продолжил свою работу и создал проект нового, более жесткого стандарта SQL2. В отличие от стандарта 1989 года, проект SQL2 предусматривал возможности, выходящие за рамки таковых, уже существующих в реальных коммерческих продуктах. А для следующего за ним стандарта SQL3 были предложены еще более глубокие изменения. В результате предложенные стандарты SQL2 и SQL3 оказались более противоречивыми, чем исходный стандарт. Стандарт SQL2 прошел процесс утверждения в ANSI и был окончательно принят в октябре 1992 года. В то время, как первый стандарт 1986 года занимает не более ста страниц, официальный стандарт SQL2 содержит около шестисот.

Вопреки стандарту SQL2, во всех существующих на сегодняшний день коммерческих продуктах поддерживаются собственные диалекты SQL. Более того, поставщики СУБД включают в свои продукты новые возможности и расширяют собственные диалекты SQL, чем еще больше отдаляют их от стандарта. Однако ядро SQL стандартизировано довольно жестко. Там, где это можно было сделать, не ущемляя интересы клиентов, поставщики СУБД привели свои продукты в соответствие со стандартом SQL-89, то же самое постепенно произойдет и с SQL2.

SQL Data Definition Language – язык определения реляционных данных

Реляционная модель данных, как и любая логическая модель данных, рассматривается с трех сторон: структура объектов, обеспечение целостности и манипулирование объектами.

Структура реляционных данных

Начнем с неформального определения терминов

Пример 1.

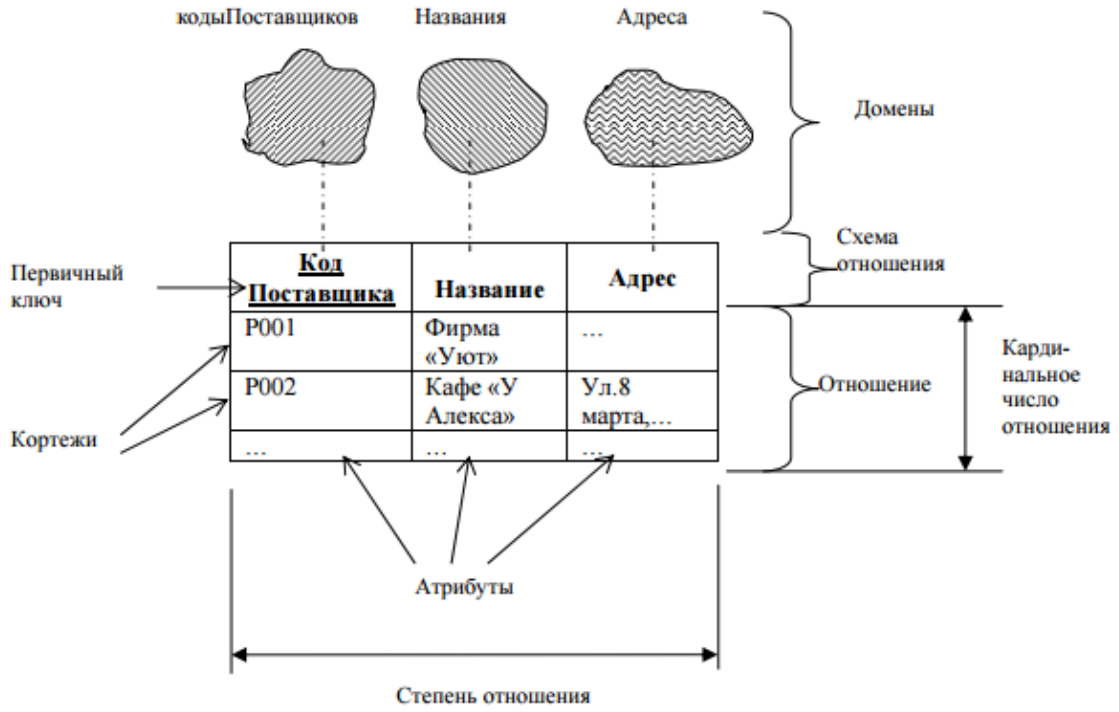

Рисунок 6.1-Реляционное отношение «ПОСТАВЩИКИ» очень подробно.

Отношение соответствует таблице, кортеж – строке таблицы, атрибут – столбцу таблицы. Первичный ключ – уникальный идентификатор для таблицы, т.е. столбец или комбинация столбцов, для которых в любой момент времени не существует двух строк с одинаковыми значениями в этих столбцах.

Домен – множество допустимых значений атрибута.

Минимальной единицей данных в реляционной модели данных является отдельное значение данных. В реляционной модели данных все отдельные значения данных являются атомарными, т.е. у них нет внутренней структуры, и они неразложимы на составляющие.

Определение реляционных данных

Домен D – конечное или счетное множество атомарных значений одного типа.

Пример 2. Домен категорий товаров, таких как «бытовая техника», «продукты питания» и т.п. может быть таким: строки русских символов длиной 20 символов.

Инструкция для создания домена в SQL (все предложения SQL даны по стандарту SQL92) выглядит так:

CREATE DOMAIN имя_домена тип_данных
DEFAULT значение;

Пример 3. Определим домен для примера 2 как

CREATE DOMAIN Category varchar(20);

Инструкция для удаления домена такова:

DROP DOMAIN имя_домена;

Инструкция SQL на создание таблицы выглядит так:

CREATE TABLE имя_таблицы (
определение_поля[, определение_поля,…n] ограничение_на_таблицу[,…n]
);

где:

определение_поля имеет вид
имя_поля тип [(размер)] [ограничение_на_поле]

ограничение_на_поле имеет вид:
[CONSTRAINT имя_ограничения]
{PRIMARY KEY | UNIQUE | NOT NULL}

ограничение_на_таблицу имеет вид:
[CONSTRAINT имя_ограничения]

{PRIMARY KEY (ключевое_поле1[, ключевое_поле2 [, ...]]) | UNIQUE (уникальное_поле1[, уникальное_поле2 [, ...]]) | NOT NULL (непустое_поле1[, непустое_поле2 [, ...]]) |
FOREIGN KEY (поле_ссылки1[, поле_ссылки2 [, ...]]) REFERENCES
внешняяТаблица [(внешнее_поле1 [, внешнее_поле2 [, ...]])]}

имя_поля, имя_таблицы – допустимые имена в Access; названия таблиц и полей, включающие пробелы, а также символы национальных кодировок, заключаются в прямоугольные скобки

«[ ]», например [название кинофильма]. Допустимыми считаются имена длиной до 64 символов, не содержащие точки «.», восклицательного знака «!», символа обратной кавычки «`» и символов квадратных скобок «[ ]», и начинающиеся не с пробела.

Инструкция SQL на удаление таблицы такова:

DROP TABLE имя_таблицы;

В стандарте SQL предложение CONSTRAINT используется для задания ограничений на значения полей записи (в частности, уникальность – UNIQUE, обязательность – NOT NULL), для установления отношений между таблицами (REFERENCES), а также для задания ограничений на таблицу целиком.

Отметим, что имена ограничений в выражениях CONSTRAINT в пределах одной базы данных должны быть различными.

Пример 4.

```
CREATE TABLE E1 (
a1 autoincrement not null CONSTRAINT pk_E1 PRIMARY KEY, a2 datetime not null,
a3 text(40)
);
```

Создается таблица из трех полей, причем поле a1 становится первичным ключом, и для этого поля создается индекс. Для определения первичного ключа здесь используется ограничение на поле с именем pk_E1.

Пример 5.

```
CREATE TABLE E2 (
b1 float,
b2 datetime, b3 long,
b4 text(100),
CONSTRAINT pk_E2 PRIMARY KEY (b2,b3), CONSTRAINT fk_E2_E1 FOREIGN KEY (b3) REFERENCES E1 (a1));
```

Создается таблица, в которой первичный ключ состоит из двух полей, b2 и b3, и при этом значения поля b3 берутся из значений поля a1 таблицы E1.

Целостность реляционных данных

В любой момент времени любая база данных содержит некоторый набор значений данных. Предполагается, что база данных является моделью реального мира. Однако, набор значений не имеет смысла, если значения не представляют определенного состояния реального мира.

Пример 6.

«Поставщик А сделал заказ на (-100) кондиционеров» - бессмысленное значение.

Следовательно, определение базы данных нуждается в расширении, включающем правила целостности. Суть правил состоит в том, чтобы сообщать СУБД об ограничениях на сочетания значений в реальном мире. Такие правила целостности называются специализированными, т.к. они применяются только в рамках конкретной базы данных.

Поэтому в реляционной модели данных есть и общие для всех реляционных баз данных правила целостности, и специализированные, относящиеся к конкретной базе данных.

Общих правил целостности всего два:

Первое правило: В первичном ключе не должно быть атрибутов, которые могут принимать неизвестные (неопределенные) значения.

В SQL для описания первичного ключа используется определение PRIMARY KEY

Второе правило: База данных не должна содержать несогласованных значений внешних ключей .

Замечание: Несогласованное значение – это значение внешнего ключа, для которого не существует отвечающего ему значения первичного ключа.

Определить внешний ключ можно оператором SQL:

FOREIGN KEY (список атрибутов внешнего ключа)
REFERENCES имя_таблицы с первичным ключом

(список атрибутов первичного ключа)

Пример 7.

Рассмотрим ситуацию, когда в базе данных предприятия хранятся сведения о сотрудниках и о детях сотрудников.

Ответим на такие вопросы:

Что произойдет, если удалить запись, на которую ссылается внешний ключ в таблице

«ДЕТИ_СОТРУДНИКОВ» (например, удалить запись о сотруднике с Таб.№ 5052, если этот сотрудник имеет двух детей)?

Что произойдет при попытке обновить значение первичного ключа записи о сотруднике, на который ссылается внешний ключ?

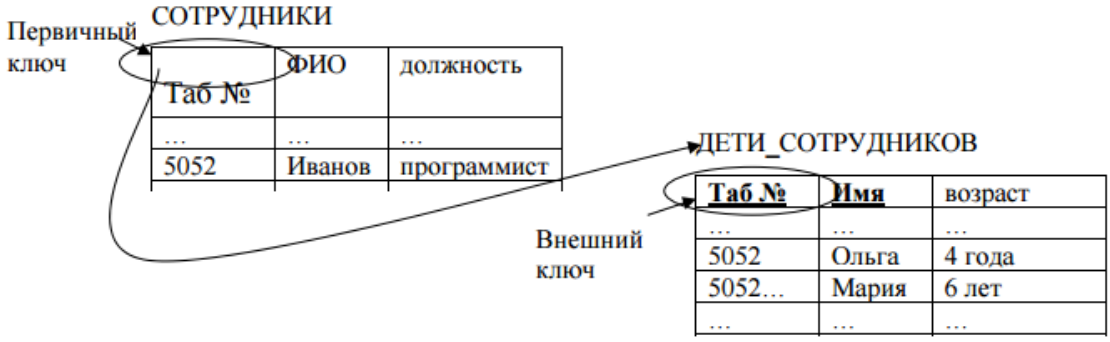

Рисунок 6.2 – Первичный и внешний ключи в таблицах

В общем, существует по крайней мере две возможности:

Ограничить (в SQL - оператор NO ACTION) –

ограничить операцию «удаление»/«обновление» до того момента, когда уже не будет существовать соответствующее данному значению Таб № первичного ключа записей в отношении с внешним ключом (в противном случае операция запрещается)

Каскадировать (в SQL - оператор CASCADE) –

каскадировать операцию «удаления»/«обновления», удаляя/обновляя также и внешние ключи.

Поэтому, в SQL92 задается ссылочная целостность в явном виде:

FOREIGN KEY (список атрибутов внешнего ключа) REFERENCES имя_таблицы

ON DELETE опция

ON UPDATE опция

где задание поведения СУБД при удалении/обновлении ключа в родительской таблице задается опцией:

CASCADE NO ACTION

вызов определенной процедуры (хранимой процедуры или процедуры триггера

Пример 8.

Вместо каскадирования или ограничения на изменение значений может вызываться некоторая процедура, например, при удалении записи информация заносится в архивную базу и т.п.

Целостность и неопределенные значения данных (null-значения)

Проблема отсутствующей информации достаточно часто встречается на практике. Для решения проблемы создаются разные подходы. Часто используются «значения по умолчанию». Например, если цвет детали не известен, или не имеет значения в данной базе, то «значение по умолчанию» - «нет» иначе – название цвета.

Другой подход, предложенный Коддом, базируется на обозначении отсутствующего значения специальным маркером, который называется null-значением. Это значение используется всегда, когда правильное значение для любого атрибута базы неизвестно или отсутствует.

Теория, изучающая null-значения, базируется на трехзначной логике (когда разрешаются три значения: «да», «нет», «UNK” – unknown - неизвестно)

Таким образом, становится очевидным правило целостности по сущностям, которые мы сформулировали в этой лекции выше:

«Ни один элемент первичного ключа не может иметь неопределенное, «null-значение»

Аналогично и для правила целостности по ссылкам:

«Всегда значения элементов внешнего ключа или являются null-значениями, или совпадают с некоторыми не null- значениями элементов первичного ключа»

Если СУБД разрешает использование null-значений, то к двум возможностям обработки ссылок по внешним ключам, добавляется третья.

Правило null-значений: Для каждого внешнего ключа разработчик должен решить, может ли внешний ключ содержать null-значения.

Как правило, можно задать разрешение на использование Null-значений на всю базу данных с помощью оператора SQL -

NULLS ALLOWED (разрешается использование Null-значений), либо

NULLS NOT ALLOWED (запрещается использование Null-значений)

Тогда, если Null-значения разрешаются, то добавляется новая опция: SET NULL (аннулировать) наряду с CASCADE и NO ACTION.

В соответствии с опцией SET NULL, при изменении записи о сотруднике внешний ключ устанавливается равным Null для всех записей о детях, а запись о сотруднике потом удаляется/обновляется.

В SQL92 к опциям CASCADE и NO ACTION добавляется SET NULL и SET DEFAULT: FOREIGN KEY (список атрибутов внешнего ключа)

REFERENCES имя_таблицы ON DELETE опция ON UPDATE опция

где опция –

либо CASCADE либо NO ACTION либо SET NULL

либо SET DEFAULT (как установлено по умолчанию для всей базы данных)

Пример 10.

Допустим, мы разрешили существование Null-значений в таблице «ДЕТИ_СОТРУДНИКОВ». Примером может быть ситуация, когда сотрудника увольняют, но о детях продолжают заботиться, и информация о них остается в базе данных.

Опишем эту ситуацию на SQL:

```
CREATE TABLE Employees
(tab_number integer NOT NULL UNIQUE, name varchar (50) NOT NULL,
position varchar (20) NOT NULL) PRIMARY KEY tab_number;
CREATE TABLE Childrens (tab_number integer NOT NULL,
child_name varchar (10) NOT NULL, age integer NOT NULL)
PRIMARY KEY (tab_number, child_name)
FOREIGN KEY (tab_number) REFERENCES Employees ON DELETE SET NULL
ON UPDATE CASCADE;
```

Таблица 1. Типы данных MS SQL Server 2000.

Тип данных	**Описание**
Bit	Один бит. В столбце с этим типом может храниться либо 0, либо 1.
Binary(n)	Двоичный данные фиксированной длины до 8000 байт.
Varbinary(n)	Двоичные данные переменной длины до 8000 байт.
Image	Двоичные данные длиной до 2Гб (2^31–1). Пространство для хранения данных этого типа отводится страницами, размер которых составляет 8 КБ.
Decimal	Используется для хранения дробных чисел с фиксированным количеством знаков до и после десятичной точки. От -10^38 +1 до 10^38 –1.
Numeric	Является аналог типа данных Decimal.
Float(n)	Используется для хранения дробных чисел с приблизительной точностью. От -1.79E+308 до 1.79E+308.
Real	Является аналогом float(24). От -3.40E+38 до 3.40E+38.
Int	Занимает 4 байта и может хранить целые числа в диапазоне от -2^31 до 2^31-1
Smallint	Занимает 2 байта и может хранить целые числа в диапазоне от -2^15 до 2^15-1
Tinyint	Занимает 1 байт и может хранить целые числа в диапазоне от 0 до 255

Bigint	Занимает 8 байт и может хранить целые числа в диапазоне от -2^63 до 2^63-1
Char(n)	Строковый тип данных фиксированной длины без поддержки Unicode длиной до 8000 символов.
Nchar(n)	Строковый тип данных фиксированной длины с поддержкой Unicode длиной до 4000 символов.
Varchar(n)	Строковый тип данных переменной длины без поддержки Unicode длиной до 8000 символов.
Nvarchar(n)	Строковый тип данных переменной длины с поддержкой Unicode длиной до 4000 символов.
Text	Строковый тип данных без поддержки Unicode длиной до 2 миллиардов символов (2^31–1). Пространство для хранения данных этого типа отводится страницами по 8 Кб.
Ntext	Строковый тип данных с поддержкой Unicode длиной до 1 миллиардов символов. Пространство для хранения данных этого типа отводится страницами по 8 Кб.
Datetime	Тип данных для хранения даты и времени с высокой точностью в диапазоне от 1 января 1753 года до 31 декабря 9999 года. Занимает 8 байт. Точность - 3,33 миллисекунды.
Smalldatetime	Тип данных для хранения даты и времени с нормальной точностью в диапазоне от 1 января 1900 года до 06 июня 2079 года. Занимает 4 байта. Точность - 1 минута.
Money	Используется для хранения денежных данных в большом диапазоне. Точность хранения до 4 знаков после десятичной точки. Занимает 8 байт. От -2^63 до 2^63 – 1.
Smallmoney	Используется для хранения денежных данных в нормальном диапазоне. Точность хранения до 4 знаков после десятичной точки. Занимает 4 байта. От -214,748.3648 до 214,748.3647.
Sql_variant	Тип данных, позволяющий хранить данные нескольких типов в одном и том же столбце. Например, при использовании этого типа данных в столбце могут находиться одновременно данные целочисленные, дробные, символьные и т.д.
Sysname	Используется для указания имен объектов СУБД. Аналог
	nvarchar(128).
Timestamp	Временной штамп. Значения в столбце этого типа изменяются SQL Server автоматически при каждом изменении строки. Можно использовать псевдоним rowversion.
Table	Этот тип данных предназначен для временного хранения сложных типов данных – для переменных Transact-SQL, параметров хранимых процедур и для значений, возвращаемых функциями пользователя.
Uniqueidentifier	Используется для хранения глобальных уникальных идентификаторов.
cursor	Предназначен для хранения ссылок на курсоры.

РАЗДЕЛ 2. ТЕХНОЛОГИИ ВЫСТАВОЧНЫХ И ЭКСПОЗИЦИОННЫХ ПРОЕКТОВ. РАЗРАБОТКА И СОЗДАНИЕ ВИРТУАЛЬНЫХ ВЫСТАВОК

Тема 8. Проекционный мэппинг в деятельности музеев

В условиях инновационного развития общества возникает необходимость использования современных креативных технологий в образовании, развитии и поддержке библиотечных и музейных специалистов нового поколения. Сегодня существует множество технологий способных вызвать огромный интерес как у специалистов различных сфер деятельности, так и у потребителей их услуг. В статье рассматривается понятийный аппарат мультимедиа технологии набирающей все большую популярность в современном обществе – проекционного мэппинга.

Проекционный мэппинг (видеомэппинг, 3D-мэппинг, в России также широко используется название «видеомаппинг») – это искусство создания и наложения проекций на физические объекты: здание, автомобиль, декорации на сцене мероприятия, одежду, часть выставочного стенда, человеческое тело (даже в динамике) и т.п., с учетом их геометрии и местоположения в пространстве.

Приставка «3D» в «3D-видеомэппинг» подразумевает, что обыгрываются свойства геометрии поверхности, на которую производится видеопроекция, а не используемую технологию, уже ставшую привычной в играх, телевизионных программах и фильмах в формате 3D. В этом контексте специальные компьютерные технологии позволяют создавать 3D-модель объекта, на который планируется осуществить проекцию, а затем изменить ее в соответствии со сценарием шоу, что дает оптическую иллюзию изменения самого объекта.

В последние несколько лет популярность таких проектов, как световые шоу на здании, а также использование 3D видеомэппинга активно растет, в связи с тем, что масштабные объемные видеопроекции и световые инсталляции очень зрелищны, а современные возможности моушн-графики и трехмерного моделирования позволяют создавать очень реалистичные, фантастические спецэффекты. Именно поэтому видеомэппинг так широко используется для проведения общественных мероприятий, городских праздников, для рекламных кампаний и презентаций.

Таким образом, проекционный мэппинг можно классифицировать по объектам, на которые осуществляется инсталляция, а именно:

- **Архитектурный мэппинг или экстерьерный мэппинг** – это трехмерная проекция на здание, мост, башню или любой другой архитектурный объект. Форма здания часто определяет контент шоу, который, как правило, обыгрывает архитектурный и исторический контекст объекта. Стены, колонны, балконы, окна, лестницы и прочие архитектурные элементы после наложения на них разнообразной графики принимают новые значения. Этот вид видеопроекции в основном используют для массовых мероприятий под открытым небом, когда нужно привлечь внимание большого количества людей.

Примером архитектурного мэппинга является 3D-mapping «Бал истории» в честь 250-летия Государственного Эрмитажа 6 декабря 2014 года на Дворцовой площади. Это представление было посвящено судьбе легендарного музея, тесно переплетенной с историей Санкт-Петербурга и России. Картины «Бала истории» – так называется проект – развернулись на фасаде здания Главного штаба, который на один вечер превратился в экран для трехмерного видеоспектакля. Зрителям напомнили о самых значительных и драматических эпизодах отечественной истории, свидетелем и непосредственным участником которых был Эрмитаж. На экране оживали события минувших эпох: царствование Екатерины II; разрушительный пожар, охвативший Зимний дворец при Николае I; строительство эрмитажных зданий и открытие первого в России публичного художественного музея… Основная идея 3D-mapping проекта – представить Эрмитаж и шедевры его коллекций как неисчерпаемый источник веры в торжество красоты и человечности.

Еще одним примером может служить 3D-мэппинг-шоу «История Дома Кондратюка», посвященное празднованию 90-летия здания Музея города Новосибирска, проходившее 21 октября 2014 года. Тема видеоинсталляции – «История Дома Кондратюка». Зрители увидели, как строился дом, какие люди в нем работали, и как в нём зародилась идея полета на Луну. Здание Музея города Новосибирска является памятником истории – «Домом Кондратюка».

- **Интерьерный мэппинг** – трёхмерная проекция производится на стены, мебель, интерьер помещений. Оптическая иллюзия позволяет волшебным образом видоизменять внутренние пространства зданий, и может служить отдельной работой, быть интегрированной в имеющееся пространство или создавать оригинальное, динамичное световое решение помещения. При этом цели могут ставиться самые различные – от создания эмоциональной атмосферы до полной трансформации пространства. Такой мэппинг отлично подходит для выставочных пространств, в частности библиотек и музеев.

Классическим примером такого мэппинга является проекционный мэппинг на выставке «Интерьерные решения», проходившей в городе Кирове 26-28 ноября 2012 года.

- **Объектный мэппинг** – в этом случае «холстом» для видеоряда служат отдельные предметы. Проецирование на малые объекты дает возможность создать на любом объекте динамичное изображение, которое создает иллюзию движения искомого объекта и нестандартного изменения его формы. Здесь мэппинг позволяет «огранить» предметы, выделить контуры, оживить детали.

Примером данного вида мэппинга в области библиотечной деятельности служит «живая» книга музея истории Екатеринбурга. Её появление в России стало первым около года назад. Подать интересно и захватывающе исторический контент – непростая задача, но медиа-книга с ней справилась на ура. Семь красочных разворотов проиллюстрировали целых два века развития и становления Екатеринбурга. Проекционное шоу предвосхищает знакомство с книгой. Объемная обложка, стилизованная под хребет Уральских гор. С помощью системы верхней проекции на белых страницах

книги проявляются анимированные сюжеты, раскрывающие ключевые исторические моменты. Все иллюстрации в издании подвижны, а текст легко воспринимается под звуковое сопровождение. Управлять мультимедийной книгой просто – привычным движением руки вы перелистываете страницу и контент меняется [3].

Подводя итоги объектной классификации стоит отметить, что все перечисленный виды проекционного мэппинга имеет широкую аудиторию зрителей, которые кроме того что смотрят и наслаждаются построенной иллюзией, также записывают видео и выкладывают в Интернет, где видеоролик просматривают еще несколько тысяч человек.

Учитывая широкое использование видеомэппинга, выделим несколько аспектов создания и реализации инсталляций таких как: выбор или изготовление поверхности на которую будет производиться проецирование; решение вопросов по созданию дизайна, световых реализаций, сюжета представлений; выбор и подготовка технических средств и программного обеспечения; поиск специалистов способных создавать и управлять функционированием всех составляющих проекционного шоу. Все это требует серьезных материальных затрат и хорошей ресурсной базы, однако результат не заставит себя долго ждать, так как видеопроекционные шоу и их эстетическое, психоэмоциональное воздействия на социум позволяет определить перспективы применения данных технологий в различных сферах деятельности современного человека.

5.2. ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Практическое занятие №1

Работа в MS ACCESS: создание реляционной базы данных

Цель работы - особенности построения реляционных баз данных, получение основных навыков работы по созданию реляционной базы данных в среде Access,

В самом широком смысле любая программа имеет дело с некоторой внешней по отношению к ее коду информацией, задающей какие-либо параметры или режим ее работы. Такую информацию также называют данными программы.

Очевидно, что в зависимости от типа решаемых задач проблемы организации работы с данными будут качественно различными. В подавляющем большинстве случаев при решении хозяйственных, экономических и финансовых задач приходится иметь дело с обширными специфически структурированными и взаимозависимыми массивами данных. Такие сложные наборы данных традиционно принято называть базами данных.

Подготовка к работе

Изучить особенности создания реляционных баз данных (материал прилагается).

Разработать структуру базы данных, содержащей справочные сведения и итоги сдачи сессий студентами одного из факультетов. В таблицах базы данных в обязательном порядке должны быть отражены следующие данные:

Таблица 1 «Студенты»: код студента (ключевое поле), фамилия, имя, отчество, номер зачетки, код группы (с использованием подстановки), дата рождения, адрес, телефон, льготы по стипендии.

Таблица 2 «Итоги сессии»: код студента (с использованием подстановки), код семестра (с использованием подстановки), код предмета (с использованием подстановки), оценка, код оценки (ключевое поле).

Таблица 3 «Предмет»: код предмета (ключевое поле), предмет.

Таблица 4 «Группа»: код группы (ключевое поле), группа.

Таблица 5 «Семестр»: код семестра (ключевое поле), курс, сессия (зимняя/летняя).

Разработать макет каждой таблицы, содержащий имя поля, тип данных, формат данных.

Ознакомиться с технологией создания таблиц в среде Access.

Изучить возможности использования «мастера подстановок» для удобства заполнения таблиц.

6. Рассмотреть возможности создания и изменения «схемы данных» для обеспечения целостности работы реляционных баз данных.

7. Ознакомиться с возможностью сортировки, фильтрации и поиска данных в режиме таблицы.

Задания на выполнение:

Создать новую базу данных, задавая свое имя базы данных.

В соответствии с разработанной структурой базы данных, заданными именами, типами и размерами полей создать в базе данных таблицы «Студенты», «Итоги сессии», «Предмет», «Группа», «Семестр».

В каждой таблице объявить указанные поля первичными ключами.

Внести данные в таблицы «Группа», «Предмет» (5 записей), «Семестр» (10 записей).

При внесении данных в таблицы «Студенты» и «Итоги сессии» предусмотреть заполнение данных с использованием подстановки значений из списка возможных значений.

Открыть схему базы данных и изменить связи между таблицами с целью обеспечения целостности данных.

Просмотреть данные таблицы «Студенты», отсортированные в алфавитном порядке.

Осуществить поиск данных в таблице «Студенты» по заданному параметру.

9. Удалить одну из записей из таблицы «Студенты», оценить изменения в других связных таблицах.

10. Результаты работы представить преподавателю.

Контрольные вопросы:

Дайте определение термина «база данных».

Какие модели данных вы знаете?

Что представляет собой реляционная модель данных?

Приведите примеры реляционных СУБД.

Каковы достоинства реляционной модели данных?

Что представляет собой база данных в системе Access?

Какие типы полей в таблицах допустимы?

Какие свойства полей вы знаете?

Что такое «первичный ключ», его назначение?

10.Что представляет собой «схема данных», ее назначение?

11. Каким образом создается схема данных из исходных таблиц базы данных?

12.Что такое «целостность данных», как ее обеспечить?

13.Организация сортировки и фильтрации данных в таблицах.

14.Организация поиска и замены данных в таблицах.

15.Возможность изменения структуры таблиц, типов и размеров полей.

16.Каково назначение «мастера подстановок»?

17.Каким образом организовать ввод значений из фиксированного списка?

Практическое занятие №2
Работа в MS ACCESS: создание форм и запросов к реляционной базе данных

Цель работы - особенности построения реляционных баз данных, получение основных навыков работы по созданию запросов и форм в среде Access.

Задания на выполнение

1. Открыть созданную вами базу данных о студентах.
2. Для каждой из существующих таблиц создать простые запросы, включающие все поля соответствующих таблиц.
3. Разработать модификации запроса к таблице «Студенты», позволяющие осуществлять просмотр в отсортированном порядке (по фамилии) осуществлять поиск по заданному полю и критерию (по фамилии и группе) осуществлять фильтрацию записей (по группе).
4. Изменить структуру одного из запросов для получения параметрического запроса.
5. Создать запрос, содержащий сведения: фамилия, имя, отчество, группа, оценка, предмет, на основе таблиц «Студенты», «Оценки».
6. Для существующих таблиц создать формы по работе с данными, используя в качестве источника информации созданные в п.2 простые запросы.
7. В каждой форме предусмотреть с помощью мастера элементов наличие элементов диалога (кнопок), позволяющих добавлять, удалять записи, закрывать форму.

Практическое занятие №3
Подготовка создание графических объектов с использованием растровой графики

В течении работы каждому студенту необходимо создать psd и jpg-файлы следующего содержимого:

1. Афишу выступления с концертом знаменитости в городе Смоленске;
2. Раскрасить черно-белое фото;
3. Воздвигнуть памятник «себе».

Практическое занятие №4
Создание аудиовизуальных произведений

В течении работы часов каждому студенту необходимо создать *.avi-файл состоящий не менее чем из 12 фрагментов с разнообразными переходами, текстом, фото, наложенными звуковыми эффектами по индивидуальному заданию.

5.3. СЕМИНАРСКИЕ ЗАНЯТИЯ

Не предусмотрены

5.4. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

График СРС

№ п/п	Название раздела (темы) дисциплины	Виды СРС	Периодичность (сроки) контроля СРС	№ семестра	Время на изучение, выполнение задания
РАЗДЕЛ 1. ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МУЗЕЙНОЙ СФЕРЕ					
1	Тема 1. История внедрения новых информационных технологий в музейное дело	Составление конспекта	1-2 недели	1	6
2	Тема 2. Современные технические средства в музейном деле	Составление конспекта; подготовка к практическому занятию	3-5 недели	1	10
3	Тема 3. Обзор и сравнительные характеристики автоматизированных информационных систем в музее	Составление конспекта; подготовка к практическому занятию	6-10 недели	1	10
4	Тема 4. Основные понятия теории баз данных	Составление конспекта; подготовка к практическому занятию	11-18 недели	1	10
				Итого в семестре	**36**
5	Тема 5. Проектирование БД методом нормализации	Составление конспекта; подготовка к практическому занятию	1-2 недели	1	10
6	Тема 6. Проектирование БД с использованием семантических моделей	Составление конспекта; подготовка к практическому занятию	3-4 недели	1	10
7	Тема 7.Язык SQL	Составление конспекта; подготовка к практическому занятию	5-6 недели	1	10
РАЗДЕЛ 2. ТЕХНОЛОГИИ ВЫСТАВОЧНЫХ И ЭКСПОЗИЦИОННЫХ ПРОЕКТОВ. РАЗРАБОТКА И СОЗДАНИЕ ВИРТУАЛЬНЫХ ВЫСТАВОК					
11	Тема 8. Интернет в музейном деле	Составление конспекта;	7-8 недели	2	10

		подготовка к практическому занятию			
12	Тема 9. Информационные технологии обработки графической информации	Составление конспекта; подготовка к практическому занятию	9-11 недели	2	10
13	Тема 10. Технологии создания обработки видеофайлов	Составление конспекта; подготовка к практическому занятию	12-14 недели	2	10
14	Тема 11. Проекционный мэппинг в деятельности музеев	Составление конспекта	15-18 недели	2	2
				Итого в семестре	**72**
				Итого по дисциплине	**108**

Курс предусматривает также самостоятельное его изучение. Формы самостоятельной работы:
- выполнение заданий, связанных с закреплением пройденного материала;
- подготовка к выполнению практических заданий;
- подготовка к контрольным работам;
- подготовка к зачетам.

Формы и методы контроля самостоятельной работы:
- проверка практических работ;

Проверка словаря и выступление на терминологическом коллоквиуме;
- тестирование;
- контрольные и проверочные работы;
- зачеты.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.

Вопросы зачета;
Тематика контрольных работ;
Задания для самостоятельной работы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Технология: объем и многозначность понятия.
- Технология как способ производства продуктов и услуг.
- Компонентная структура технологии.
- Атрибутивные признаки и критерии технологии.
- Технологическая система: понятие, структура.
- Информационная технология: объем и содержание понятия.
- Компонентная структура информационной технологии (цель, предметы).
- Компонентная структура информационной технологии (ресурсы и средства).
- Компонентная структура информационной технологии (регламенты, процессы, продукты/услуги).
- Тенденции развития информационных технологий.
- Табличные процессоры. Понятие, цели и задачи.
- Функции табличных процессоров.
- Достоинства и недостатки существующих систем.
- Решаемые задачи.
- Работа с диаграммами. Виды.
- Спарклайны и фильтры.
- Работа с формулами в табличных процессорах.

- Получение внешних данных.
- Сортировка и фильтр. Работа с данными.
- Возможности работы с ячейками.
- Использование ИТ для системного описания информационного производства.
- Специфическая цель информационной технологии.
- Технологическая структура информационного производства
- Номенклатура и содержание процессов информационной технологии.
- Структура ресурсов информационной технологии.
- Специфические обеспечивающие средства ИТ.
- Регламентирующая документация.
- Производственная структура.
- Информационные продукты и услуги.
- Технология. Понятие технологии.
- Основатель технологии, как отдельной дисциплины.
- Составляющие технологии.
- Методологическая основа технологии.
- Признаки научно и практически обоснованной технологии.
- Понятие современной технологии.
- Определение информационной технологии.
- Автоматизированная информационная технология.
- Этапы развития информационных технологий.
- Вид задач и процессов обработки информации.
- Проблемы, стоящие на пути информатизации общества.
- Преимущество, которое приносит компьютерная технология.
- Классификация информационных технологий способу реализации в автоматизированных информационных системах.
- Классификация информационных технологий по степени охвата АИТ задач управления.
- Классификация информационных технологий по классам реализуемых технологических операций.
- Классификация информационных технологий по типу пользовательского интерфейса.
- Классификация информационных технологий по вариантам использования сети ЭВМ.
- Классификация информационных технологий по обслуживаемой предметной области.
- Феномен информатизации современного общества.
- Этические и социальные проблемы информатизации.
- Основы теории информации и информационной технологии.
- Информатизация в социальной политике и управлении социальной сферой.
- Зарубежный опыт информатизации социальной работы.
- Информационные технологии дистанционного обучения.
- Практическое использование информационных технологий в социальной сфере.
- Интегрированные информационные системы в социальной сфере.
- Программные средства автоматизации управленческой деятельности в социальной сфере в России.
- История информационных технологий.
- Основные понятия реляционных баз данных.
- Структура и функционал языка гипертекстовой разметки HTML.
- Понятия гипермедиа и мультимедиа: многосторонность понятий и подходов.
- Современные языки программирования.
- Состояние рынка телекоммуникационных технологий.
- Назначение экспертных систем.
- Рынок справочно-информационных систем.
- Автоматизированные системы обработки информации и управления.
- Примеры систем организационного управления.
- Системы дистанционного образования в России.
- Направлений использования ИТ в культуре и искусстве.
- Электронная библиотека e-Library.
- Основные возможности системы «Моя библиотека».

- Функционал Microsoft Excel.

ВОПРОСЫ ЗАЧЕТА

1. Технология: объем и многозначность понятия. Технология как способ производства продуктов и услуг.
2. Компонентная структура технологии. Атрибутивные признаки и критерии технологии. Технологическая система: понятие, структура.
3. Технология как наука (научное описание способов производства). Признаки технологии как научной дисциплины: предмет и задачи изучения, понятийный аппарат, научные принципы и законы, исследовательские методы. Прикладной характер технологического знания.
4. Информационная технология: объем и содержание понятия. Компонентная структура информационной технологии (цель, предметы, ресурсы и средства, регламенты, процессы, продукты/услуги).
5. История и тенденции развития информационных технологий.
6. Классификация информационных технологий (базовые, прикладные, специальные).
7. Технологии баз данных.
8. Гипертекстовые технологии.
9. Гипермедиа технологии. Мультимедийные технологии.
10. Технологии программирования. Телекоммуникационные (сетевые) технологии. Искусственный интеллект.
11. Информационные технологии в политике и социальной сфере.
12. Информационные технологии в промышленности и экономике.
13. Информационные технологии автоматизированного проектирования.
14. Информационные технологии организационного управления.
15. Информационные технологии в образовании.
16. Информационные технологии в культуре и искусстве.
17. Библиотечные технологии.
18. Библиографические технологии.
19. Офисные технологии. Образовательные технологии.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИН

7.1.РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

7.1.1.ОСНОВНАЯ ЛИТЕРАТУРА

1. Алешин, Л.И. Автоматизация в библиотеке: учебное пособие. Часть 1/Л.И. Алешин. – М.: МГУКИ, ИПО Профиздат, 2002. – 176 с. – (Современная библиотека). – Вып. 14.
2. Алешин, Л.И. Автоматизация в библиотеке: учебное пособие. Часть 2/Л.И. Алешин. – М.: МГУКИ, ИПО Профиздат, 2001. – 144 с. – (Современная библиотека). – Вып. 14.
3. Алешин, Л.И. Телекоммуникационные технологии для библиотек: учебное пособие/Л.И. Алешин. – М.: Литера, 2009. – 352 с. – (Современная библиотека). – Вып. 56.
4. Левин, М. Компьютерные сети: устройство, подключение и использование/М. Левин. – М.: Оверлей, 2001. – 416 с.
5. Денисов, А. Интернет: самоучитель/ А. Денисов, И.Вихорев, А. Белов. – СПб: Питер, 2001. – 464с.:ил.
6. Пилко, И.С. Информационные и библиотечные технологии: учебное пособие/И.С. Пилко. – СПб.: Профессия, 2006. – 342 с. – (Библиотека).

7.1.2. ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Антопольский А.Б. Информационные ресурсы России [Текст] : науч.-метод. пособие / А.Б. Антопольский. – М. : Либерея, 2004. – 423 с.
2. Брежнева, В.В. Информационное обслуживание: продукты и услуги, предоставляемые библиотеками и службами информации предприятий [Текст] : учеб-практ. пособие / В.В. Брежнева, В.А. Минкина; СПбГУКИ. – СПб. : Профессия, 2004. – 303 с.
3. Вендров В.Л. CASE-технологии., Современные методы и средства проектирования ИС. – М.: Финансы и статистика, 2002. 176 с.
4. Вохрышева Е.В. Медиатехнологии - путь в будущее современных библиотек [Текст] : учеб.-практ. пособие. – М. : Либерея-Бибинформ, 2005. - 142 с.
5. Гринберг, А.С. Информационные технологии управления [Текст] : учеб. пособие / А.С. Гринберг, Н.Н. Горбачев, А.С. Бондаренко. – М.: ЮНИТИ-ДАНА, 2004. – 479 с.
6. Информатика как наука об информации: информ, док., технолог., экон, социальн. и орг. аспект [Текст] /Под ред Р.С. Гиляревского. - М. : ФАИР-ПРЕСС, 2006. – 592 с.

7. Кривцов А.Н,, Лёзина Т.А. и др. Информационные системы в экономике. Учебное пособие. – СПб, 2005 г.
8. Советов, Б.Я. Информационные технологии [Текст] : учебник / Б.Я. Советов, В.В. Цехановский. – М. : Высш. шк., 2003. – 263 с.
9. Шрайберг, Я.Л.Основные положения и принципы разработки автоматизированных библиотечно-информационных систем и сетей [Текст] : Учеб.-практ. пособие / Я.Л. Шрайберг. – Изд- е 2-е, испр. и доп. – М.: Либерея, 2001. – 103 с.

7.2 СРЕДСТВА ОБЕСПЕЧЕНИЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

7.2.1 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ И МАТЕРИАЛЫ ПО ВИДАМ ЗАНЯТИЙ

Скачков Д.Е. Информационные технологии: методические указания к выполнению практических работ по дисциплине "Информационные технологии" для специальностей 070503.65 "Музейное дело и охрана памятников", 071201.65 "Библиотечно-информационная деятельность" / Д.Е.Скачков. – Смоленск: СГИИ, 2012. – 53 с

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ КУРСА ДЛЯ ПРЕПОДАВАТЕЛЯ

При подготовке к занятию преподаватель, ведущий семинарские занятия, должен внимательно ознакомиться с учебно-методическим комплексом по дисциплине и уточнить план его проведения. Более того необходимо ознакомиться с новыми публикациями по теме семинара и дополнительной литературы по вопросам плана занятия. Оказывать методическую помощь студентам в подготовке докладов и рефератов. В ходе семинара во вступительном слове преподаватель должен раскрыть теоретическую и практическую значимость темы семинарского занятия, определить порядок его проведения, время на обсуждение каждого учебного вопроса. Дать возможность выступить всем желающим, а также предложить выступить тем студентам, которые по тем или иным причинам пропустили лекционное занятие или проявляют пассивность. Целесообразно в ходе обсуждения учебных вопросов задавать выступающим и аудитории дополнительные и уточняющие вопросы с целью выяснения их позиций по существу обсуждаемых проблем. Поощрять выступления с места в виде кратких дополнений и постановки вопросов выступающим и преподавателю. Для наглядности и закрепления изучаемого материала преподаватель может использовать таблицы, схемы и т.д. В заключительной части семинарского занятия следует подвести его итоги: дать объективную оценку выступлений каждого студента и учебной группы в целом. Раскрыть положительные стороны и недостатки проведенного семинарского занятия. Ответить на вопросы студентов. Назвать тему очередного занятия. К тому же преподаватель обязан проводить групповые и индивидуальные консультации студентов по вопросам, возникающим у студентов в ходе их подготовки к текущей и промежуточной аттестации, рекомендовать в помощь учебные и другие материалы, а также справочную литературу. Освоение дисциплины предполагает использование как традиционных (лекции, практические занятия с использованием методических материалов), так и инновационных образовательных технологий с использованием в учебном процессе интерактивных форм проведения занятий. Семинарские занятия в традиционной форме проводятся в соответствии с утвержденной учебно-методической разработкой для проведения семинарских занятий и самостоятельной работы студентов по соответствующей дисциплине. Каждая тема разработки содержит перечень основных вопросов для обсуждения, контрольные вопросы темы, задания для самостоятельной работы и список литературы, рекомендуемой к изучению. Если занятие проводится в интерактивной форме, преподаватель должен обсудить в группе на предыдущем занятии план проведения последующего семинара в интерактивной форме. При этом необходимо распределить задания, темы презентаций, дать рекомендации студентам по подготовке и проведению занятия в интерактивной форме. В процессе обучения необходимо обращать внимание в первую очередь на те методы, при которых слушатели идентифицируют себя с учебным материалом, включаются в изучаемую ситуацию, побуждаются к активным действиям, переживают состояние успеха и соответственно мотивируют свое поведение. Всем этим требованиям в

наибольшей степени отвечают интерактивные методы обучения. Учебный процесс, опирающийся на использование интерактивных методов обучения, организуется с учетом включенности в процесс познания всех студентов группы без исключения. Совместная деятельность означает, что каждый вносит свой особый индивидуальный вклад, в ходе работы идет обмен знаниями, идеями, способами деятельности. Организуются индивидуальная, парная и групповая работа, используется проектная работа, осуществляется работа с документами и различными источниками информации. Интерактивные методы основаны на принципах взаимодействия, активности обучаемых, опоре на групповой опыт, обязательной обратной связи. Создается среда образовательного общения, которая характеризуется открытостью, взаимодействием участников, равенством их аргументов, накоплением совместного знания, возможностью взаимной оценки и контроля. Ведущий преподаватель вместе с новыми знаниями ведет участников обучения к самостоятельному поиску. Активность преподавателя уступает место активности студентов, его задачей становится создание условий для их инициативы. Преподаватель отказывается от роли своеобразного фильтра, пропускающего через себя учебную информацию, и выполняет функцию помощника в работе, одного из источников информации.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ КУРСА ДЛЯ ОБУЧАЮЩЕГОСЯ

Успешное освоение материала курса возможно лишь при систематической работе в соответствии с РПД. Поможет в этом и серьезное изучение ряда базовых дисциплин Курс дает материал обобщающего характера, он конкретизируется в других дисциплинах. Для овладения понятийным аппаратом важно не только знание терминологических стандартов, но их оценка специалистами. Важно отслеживать изменения в них по печатным и электронным источникам. Освоение сложного курса в последнее время облегчено изданием учебной литературы разных жанров. Она указана в списках основной и дополнительной литературы. Вместе с тем, разнообразие научных концепций и подходов к содержанию учебной дисциплины усиливает ориентирующую роль лекций и занятий, проводимых преподавателем. При подготовке к любым видам занятий, читая и конспектируя источники, необходимо выделять спорные моменты, противоположные точки зрения и др. Самостоятельная работа, как аудиторная, так и внеаудиторная, осуществляется в виде подготовки к семинарам и практическим занятиям, выполнения домашних заданий. Обязательно использование новых информационных технологий: поиск определений в сети, мониторинг отраслевого документального потока. При изучении курса необходимы:

1. подготовка к семинарским занятиям;
2. подготовка к практическим занятиям;
3. создание собственного «банка данных», включающего:

а) дайджесты к семинарам;

б) «тезаурус» (словарь основных терминов предметной области);

в) структурно-логические схемы и таблицы;

г) выполнение аудиторных практических занятий и домашних заданий.

Подготовка к лекции не нужна. Подготовка к семинарским занятиям заключается в освоении теоретического материала по рекомендуемой литературе и конспектам лекций. После выполнения задания обсуждаются результаты. Подготовка к зачету должна быть регулярной. Она начинается с первого занятия (общее знакомство с ресурсной базой дисциплины, в том числе ее методическим обеспечением; информирование о формах контроля) и завершается подготовкой к тестированию - повторением материала дисциплины

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ ОБУЧАЮЩИХСЯ

Обучающийся, в ходе выполнения самостоятельной работы должен проявить способность к самостоятельному поиску в русле выбранной проблематики; умение находить и использовать нужную информацию; показать умение строить научное развернутое и аргументированное высказывание.

При изучении материала необходимо наличие требуемых текстов для рассмотрения. Для достижения четкости и структурированности работы студент должен фиксировать выполнение самостоятельных заданий и оформлять записи в рабочих тетрадях.

Для успешного освоения материала профессиональных статей и составления грамотного конспекта необходимо сначала внимательно прочитать статью или все статьи, выделить основные положения и только после этого приступить к конспектированию. Конспект не должен превращаться в механическое «переписывание», в конспекте нужно кратко и сжато отразить основные концепции статьи. Самый лучший конспект – тезисы, которые являются результатом глубокой проработки материала.

Одной из важных форм повышения качества подготовки студентов является чётко организованная самостоятельная работа, которая протекает во всех видах учебной работы и в самых различных формах. Задача преподавателя состоит в том, чтобы активизировать эту деятельность у студентов и управлять ею.

Одной из форм самостоятельной работы студентов является подготовка к семинарским занятиям. Она включает в себя: отработку лекционного материала, подготовку вопросов, вынесенных на самостоятельное изучение, проработку рекомендованной литературы, конспектирование материалов учебников и статей.

Обсуждение вопросов на семинарском занятии позволит студентам научиться чётко и логично излагать свои мысли, вести дискуссию, аргументировано защищать свою точку зрения.

В рамках изучения курса «ИТ в музейной работе» особым видом самостоятельной работы является работа творческого характера. Она может быть представлена в форме исследований, докладов, рефератов, контрольных работ. Эта работа носит особый характер, она помогает овладеть методами поисковой деятельности, обработки, обобщения, анализа информации. Она способствует проявлению индивидуальности, формированию творческого мышления, активизации интеллектуального потенциала, углублению и закреплению полученных при обучении знаний, расширению общего кругозора.

Каждая тема курса сопровождается списком обязательной и дополнительной литературы, а также вопросами и заданиями для самостоятельной работы. Студент должен овладеть умениями и навыками самостоятельной работы с предлагаемыми источниками, продемонстрировать способность самостоятельно сформулировать проблему и правильно интерпретируя материал делать собственные выводы.

При изучении курса «ИТ в музейной работе» процесс развития самостоятельной работы студентов должен происходить систематически, что будет способствовать формированию способности к самообразования, без которого выпускник не сможет стать конкурентоспособным специалистом

7.2.2. ИНФОРМАЦИОННО-ПРОГРАММНЫЕ СРЕДСТВА

1. Персональные компьютеры с доступом к сети Интернет.
2. Программное обеспечение: пакет программ Microsoft Office (MS Word, MS Excel, MS PowerPoint, MS Access), браузеры (Chrome, IE, Яндекс браузер).

Интернет-ресурсы:

- CITforum – http://www.citforum.ru/
- Электронная библиотека материалов в области информационно-коммуникационных технологий – http://www.ict.edu.ru/lib
- Институт систем информатики им. А.П.Ершова (ИСИ СО РАН) – http://www.iis.nsk.su/
- Современные технологии автоматизации – http://www.cta.ru/
- Афинотека – http://athena.vvsu.ru/
- Библиотека братьев Фроловых – http://www.frolov-lib.ru/
- Библиотека электронных книг по информатике и компьютерным технологиям – http://eruditus.name/kopilka.html
- Бесплатная электронная библиотека Чувашского государственного университета им. И. Н. Ульянова – http://by-chgu.ru/category/informatika

- Перспективные информационные технологии и интеллектуальные системы – http://pitis.tsure.ru/
- WwwMASTER.ru – http://www.wwwmaster.ru/
- Библиотека учебников компьютерной тематики – http://biblioteka.net.ru/
- Библиотека компьютерной литературы – http://www.netbook.perm.ru/komputer.html

.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Специализированные аудитории. При изучении дисциплины используются аудитории, оборудованные мультимедийными средствами обучения: проектором, ноутбуком, интерактивной доской. Использование интернет-ресурсов предполагает проведение занятий в компьютерных классах с выходом в Интернет. В компьютерных классах обучающиеся имеют доступ к информационным ресурсам, к базе данных библиотеки.

Для обучающихся с ограниченными возможностями здоровья необходимы специальные условия для получения образования. В целях доступности получения высшего образования по образовательным программам инвалидами и лицами с ограниченными возможностями здоровья Институтом обеспечивается: 1. Наличие альтернативной версии официального сайта Института в сети «Интернет» для слабовидящих; 2. Присутствие ассистента, оказывающего обучающемуся необходимую помощь; 3.Для инвалидов и лиц с ограниченными возможностями здоровья по слуху – дублирование вслух справочной информации о расписании учебных занятий; обеспечение надлежащими звуковыми средствами воспроизведения информации; 4. Для инвалидов и лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, созданы материально-технические условия обеспечивающие возможность беспрепятственного доступа обучающихся в учебные помещения, объекту питания, туалетные и другие помещения Института, а также пребывания в указанных помещениях (наличие расширенных дверных проемов, поручней и других приспособлений).

Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

8.2. Учебно-лабораторное оборудование

Помещения для самостоятельной работы обучающихся должны быть оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

передачу видео.

ЛИСТ СОГЛАСОВАНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа дисциплины ***Информационные технологии в музейной работе*** разработана в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению

51.03.04 Музеология и охрана объектов культурного и природного наследия, направленности ***Культурный туризм и экскурсионная деятельность. Преподавание специальных дисциплин*** (утвержден ***Приказом Минобрнауки России № 1180 от 06.12.2017 г. с изменениями и дополнениями от 26.11.2020 г., 08.02.2021г.***), учебным планом института по этому же направлению, утвержденным Ученым советом ***24.06.2021г., Протокол № 7***, с учетом основной профессиональной образовательной программы (утверждена ***24.06.2021 г.***).

Автор программы – ***Скачков Д.Е., ст. преподаватель***

Рабочая программа рассмотрена и одобрена на заседании кафедры ***библиотечно-информационной деятельности и музеологии*** (рецензент – ***_____,*** протокол №___ от «____» _____ 20___г.)

Заведующий кафедрой _____ Е.С. Мертенс _____
(подпись) (И.О.Фамилия) (дата)

Согласовано:
Начальник УМУ _____ В.В.Азарова _____