

**КОМПЬЮТЕРНЫЕ
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**
Раздел 2. Технологии баз данных и знаний

Краткий конспект лекций

Выдержки из учебного пособия:

Компьютерные информационные технологии: Учеб. пособие для иностранных студентов экономических специальностей / М. Н. Садовская, В. С. Оскерко, Л. К. Голенда, Н. Н. Говядинова, – Минск.: БГЭУ, 2015. – 299 с.

Минск

ОГЛАВЛЕНИЕ

11. ОРГАНИЗАЦИЯ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ.....	4
11.1. Экономическая информация, ее виды и структурные единицы	4
11.2. Внемашинная организация экономической информации.....	5
11.3. Внутримашинная организация экономической информации.....	8
Глоссарий	10
12. МОДЕЛИ ДАННЫХ.....	12
12.1. Трехуровневая модель организации баз данных	12
12.2. Иерархическая модель	12
12.3. Сетевая модель	13
12.4. Реляционная модель.....	13
12.5. Постреляционная модель	15
12.6. Объектно-ориентированная и объектно-реляционная модели.....	16
12.7. Многомерная модель	17
Глоссарий	18
13. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	20
13.1. Требования, предъявляемые к БД	20
13.2. Этапы жизненного цикла БД.....	20
13.3. Модель «сущность-связь»	20
13.4. Преобразование ER-модели в реляционную модель.....	24
13.5. Нормализация таблиц	25
13.6. Этапы проектирования базы данных и их процедуры	26
13.7. Семантическая объектная модель	27
13.8. CASE-средства для моделирования данных	29
Глоссарий	30
14. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫМИ.....	33
14.1. Понятие СУБД.....	33
14.2. Архитектура СУБД	33
14.3. Классификация СУБД.....	34
14.4. Возможности, предоставляемые СУБД пользователям	34
14.5. Производительность СУБД.....	35
14.6. Режимы работы пользователя в СУБД	35
14.7. Функции СУБД.....	36
14.8. Направления развития СУБД.....	36
Глоссарий	36
15. ОБЩАЯ ХАРАКТЕРИСТИКА СУБД MICROSOFT ACCESS 2010.....	38
15.1 Характеристики Microsoft Access 2010	38
15.2. Характеристика объектов базы данных	38
15.3. Пользовательский интерфейс	40
15.4. Настройка рабочей среды.....	41

15.5. Типы данных и выражения.....	42
Глоссарий	44
16. ВВЕДЕНИЕ В ЯЗЫК SQL	46
16.1. Язык SQL в СУБД.....	46
16.2. Структура команды, типы данных и выражения в SQL	46
16.3. Функциональные возможности языка SQL.....	47
16.4. Диалекты SQL в СУБД.....	50
Глоссарий	50
17. СИСТЕМЫ ОБРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ БД.....	51
17.1. Эволюция концепций обработки данных	51
17.2. Системы совместного использования файлов	52
17.3. Клиент/серверные системы.....	53
17.4. Системы обработки распределенных БД.....	55
17.5. Типы интерфейсов доступа к данным базы	58
17.6. Хранилище данных	58
Глоссарий	61
18. АДМИНИСТРИРОВАНИЕ БАЗЫ ДАННЫХ.....	63
18.1. Функции администратора базы данных.....	63
18.2. Защита баз данных	64
18.3. Правовая охрана баз данных.....	65
18.4. Восстановление базы данных	65
18.5. Оптимизация работы базы данных.....	66
18.6. Возможности администрирования БД в СУБД Access	68
Глоссарий	68
19. БАЗЫ ЗНАНИЙ И МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ.....	70
19.1. Базы знаний.....	70
19.2. Продукционные модели.....	71
19.3. Семантические сети	72
19.4. Фреймы.....	73
19.5. Формальные логические модели	74
Глоссарий	76
20. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	78
20.1. Задания по теме «Моделирование базы данных в программе ERwin» ..	78
20.2. Задания по теме «Технологии работы с базой данных	81
в СУБД Microsoft Access»	81
ЛИТЕРАТУРА	88

11. ОРГАНИЗАЦИЯ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ

11.1. Экономическая информация, ее виды и структурные единицы

Экономическая информация (ЭИ) – это совокупность различных сведений экономического характера, используемых для планирования, учета, контроля, анализа и управления народным хозяйством и его звеньями. ЭИ включает сведения о трудовых, материальных и денежных ресурсах и деятельности экономических объектов (предприятий, организаций, банков, фирм и т.д.) на определенный момент времени. Эти сведения представляются натуральными и стоимостными показателями. ЭИ можно классифицировать по:

- *функциям управления* – учетная, плановая, статистическая и др.;
- *месту возникновения* – внутренняя и внешняя;
- *стадиям образования* – первичная и вторичная;
- *способу представления* – цифровая, алфавитно-цифровая, графическая;
- *стабильности* – переменная, условно-постоянная, постоянная;
- *полноте* – недостаточная, достаточная, избыточная;
- *истинности* – достоверная, недостоверная;
- *временному периоду возникновения* – периодическая и непериодическая.

Структурными единицами ЭИ являются реквизиты, показатели, документы, массивы. **Реквизит** – это характеристика свойства объекта. Реквизиты подразделяются на реквизиты-признаки и реквизиты-основания. *Реквизит-признак* (РП) характеризует качественные свойства объекта, например, Ф.И.О. исполнителя. *Реквизит-основание* (РО) дает количественную характеристику объекта, выраженную в определенных единицах измерения, например, цена продукта в рублях. Реквизиты имеют наименования и значения. Область значений описывается форматом. *Формат* определяет тип и максимальную длину значений. Тип может быть числовым, символьным, логическим и дата/время. Для записи формата используются определенные символы. Так цифра 9 в формате означает

позицию десятичной цифры, А – буквы. Примеры записи форматов: 9(3) – это означает, что значения реквизита являются десятичными целыми числами максимум трехзначными; А(10) – текстовыми, содержащими максимум 10 букв.

Показатель – это совокупность реквизита-основания и логически связанных с ним реквизитов-признаков, имеющих экономический смысл. Так, в табл.11.1 содержится показатель – «Количество изделий, изготовленное работником цеха № 7 Макаровым И.И., составило 50 шт.».

Таблица 11.1

Пример показателя

Цех	Ф.И.О.	Количество изделий, шт.
7	Макаров И. И.	50
РП ₁	РП ₂	РО

На основе показателей строятся документы. **Документ** – материальный объект, содержащий информацию, оформленную в установленном порядке, и имеющий правовое значение в соответствии с действующим законодательством. Экономические объекты широко применяют различные документы (акты, сводки, ведомости и др.) для отражения своей деятельности.

Массив – это совокупность документов одного вида, объединенных по определенному признаку, например, множество финансовых отчетов предприятий некоторой отрасли.

11.2. Внемашинная организация экономической информации

Для повышения эффективности работы с информацией разрабатываются автоматизированные информационные системы (АИС). Их важнейший элемент – информационное обеспечение. Оно включает, в частности, информационную базу, хранящуюся на различных носителях, и состоящую из двух взаимосвязанных частей – внемашинной и внутримашинной. К *внемашинной* относится та часть ЭИ, которая обслуживает АИС в виде, воспринимаемом человеком без каких-либо технических средств. Она представлена различными документами, которые можно классифицировать по:

- *сфере деятельности* – статистические, банковские, финансовые и др.;
- *отношению к объекту управления* – входящие, исходящие, промежуточные, архивные;
- *содержанию хозяйственных операций* – материальные, денежные, расчетные;
- *назначению* – распорядительные, исполнительные, комбинированные;
- *способу использования* – разовые и накопительные;
- *способу заполнения* – заполняемые вручную или при помощи технических средств.

Развитие АИС потребовало унификации и стандартизации документов, которые предписывают выделять в документе:

- *заголовочную часть*. Она включает: наименование объекта, характеристику документа (индекс), наименование документа и др.;
- *содержательную часть*. Она представлена обычно в виде таблицы, где располагаются показатели;
- *оформляющую часть*. Она содержит подписи лиц, ответственных за правильность его составления, а также дату составления.

Для представления информации, содержащейся в документах, в форме, удобной для ввода и обработки данных с помощью компьютеров, производится классификация и кодирование информации.

Классификация – это распределение множества объектов на подмножества в соответствии с установленными признаками сходства или различия. Признак сходства или различия, положенный в основу классификации, называется ее *основанием*. Совокупность правил классификации и результат классификации называется *системой классификации*. Существует две системы классификации – иерархическая и фасетная. В *фасетной* системе заданное множество объектов делится на группировки по нескольким независимым признакам (фасетам). Например, промышленную продукцию «Обувь» можно классифицировать: по материалу – на резиновую, кожаную, матерчатую и др.; по качеству – на модельную, рядовую; по половозрастному признаку – на мужскую, жен-

скую, детскую. *Иерархическая* система применяется в случае, когда какое-либо множество объектов подразделяется на классы, подклассы, группы и т.д. последовательно по взаимоподчиненным основаниям. Пример классификации по такой системе: ресурсы → материальные, людские; материальные → земля, капитал; людские → труд, предпринимательская способность.

Кодирование информации – это образование и присвоение кодового обозначения объекту классификации, признаку классификации и (или) классификационной группировке. Коды позволяют: уменьшить объем информации, вводимой в ЭВМ; облегчить запись на машинные носители; осуществить поиск и сортировку; обеспечить наглядность выходных документов. Их классифицируют по: *форме представления* – цифровые и алфавитно-цифровые; *длине* – однозначные и многозначные; *методу образования* – порядковые, серийно-порядковые, разрядные, комбинированные. В случае *порядкового метода* кодирования объектам присваиваются порядковые номера, начиная с единицы. *Серийно-порядковый* метод предполагает выделение серии номеров, а внутри серии – присвоение порядковых номеров (например, нумерация комнат на этаже). *Разрядный метод* применяется для кодирования объектов, определяемых несколькими соподчиненными признаками, каждому из которых отводится определенное число разрядов, в пределах которого кодирование начинается с единицы.

Коды в документах проставляются согласно классификаторам. **Классификатор** – это систематизированный свод наименований объектов, признаков классификации и их кодовых обозначений. Он используется для ручного проставления кодов в первичных документах в ходе подготовки их к компьютерной обработке и автоматического декодирования информации и формирования необходимых текстов в выходных документах. Существуют следующие виды классификаторов: *общегосударственные* (например, классификатор отраслей народного хозяйства), *отраслевые* (например, классификатор видов оплат и удержаний из заработной платы в пищевой промышленности), *локальные* (например, классификатор структурных подразделений предприятия).

11.3. Внутримашинная организация экономической информации

Внутримашинная часть информационной базы АИС хранится на машинных носителях в отдельных не связанных между собой файлах данных или в едином месте, называемом базой данных. Недостатки файловой организации, необходимость централизации данных с целью коллективного доступа к ним, высокие требования к скорости обработки и достоверности данных стали причинами, обусловившими появление и быстрое развитие баз данных. Базу данных в общем смысле можно определить как определенным образом организованную совокупность данных. Тогда база данных и справочник, и словарь, и энциклопедия, и телефонная книга.

Концепцию баз данных разработала американская компания IBM в 1970 г. Согласно ей *база данных* (БД) – это именованная совокупность данных, отображающая состояние объектов, их свойства и взаимоотношения в некоторой предметной области. *Предметная область* – это часть реального мира, которая описывается и моделируется с помощью базы данных. *Объектом* может быть предмет, вещество, событие, лицо, явление, абстрактное понятие, то есть все то, что может характеризоваться набором значений некоторой совокупности атрибутов. *Атрибут* – это информационное отображение свойства объекта. Например, объект «книга» характеризуется атрибутами: «наименование», «авторы», «количество страниц», «тираж», «цена» и др.

БД можно рассматривать как *информационную модель объекта*, от обоснованности, точности и достоверности которой во многом зависит эффективность управления объектом. В БД *информация хранится централизованно*. Многие могут иметь возможность доступа, просмотра и изменения данных в одно и то же время, при этом пользуясь самой последней версией информации. Централизованное хранение позволяет легче изменять данные, согласовывать их, экономить дисковое пространство. Организация данных в БД дает *возможность быстрого поиска, анализа и отображения данных*. БД представляет собой новый подход к организации данных по сравнению с файловой организаци-

ей. Он состоит в том, что возможно *обращение к данным без знания физического расположения их в памяти компьютера*, вследствие чего доступ к данным и их обработка более просты. В этом главные преимущества БД над файловой организацией данных.

Приложениями БД являются:

- запросы. **Запрос** – это требование пользователя на отбор данных из базы и/или на выполнение определенных действий;
- формы. Формы используются для разных целей. Наиболее часто они применяются для ввода, просмотра и редактирования данных;
- отчеты. **Отчет** – это представление информации из БД в виде, удобном для ее восприятия и анализа пользователем;
- Web-страницы, предназначенные для публикации БД в Интернет;
- прикладные программы. Данными, входящими в состав БД, управляет программная система, называемая системой управления базами данных (СУБД).

Прикладная программа – это программа, работающая с БД, и написанная на языке программирования, встроенном в СУБД.

БД состоит из четырех основных *компонентов*:

- 1) *данных пользователей*;
- 2) *метаданных*. Это описание структуры БД, произведенное СУБД на основании пользовательского описания (*словарь данных*);
- 3) *данных, призванных улучшить производительность и доступность БД*. Они состоят, главным образом, из индексов (их аналог – шифры книг, ускоряющие их поиск);
- 4) *метаданных приложений*. Это описания структуры и формата пользовательских запросов, форм, отчетов и других приложений, выполненное СУБД.

Пользователь БД – это лицо или прикладная программа, которые могут обращаться с командами и/или запросами к БД и получать от нее результаты обращений. Люди, работающие с БД, – это конечные пользователи БД (специалисты предметной области, которым требуется информация из БД для выпол-

нения служебных обязанностей) и обслуживающий персонал (администраторы БД и разработчики прикладных программ).

Глоссарий

Атрибут – информационное отображение свойства объекта.

База данных – именованная совокупность данных, отображающая состояние объектов, их свойства и взаимоотношения в некоторой предметной области.

Документ – материальный объект, содержащий информацию, оформленную в установленном порядке, и имеющий правовое значение в соответствии с действующим законодательством.

Запрос – требование пользователя на отбор данных из базы и/или на выполнение определенных действий над ними.

Классификация информации – распределение множества объектов на подмножества в соответствии с установленными признаками сходства или различия.

Кодирование информации – образование и присвоение кодового обозначения объекту классификации, признаку классификации и/или классификационной группировке.

Массив – совокупность документов одного вида, объединенных по определенному признаку.

Объект – предмет, вещество, событие, лицо, явление, абстрактное понятие, то есть все то, что может характеризоваться набором значений некоторой совокупности атрибутов.

Отчет – представление информации из БД в виде, удобном для ее восприятия и анализа пользователем.

Показатель – совокупность реквизита-основания и логически связанных с ним реквизитов-признаков, имеющих экономический смысл.

Пользователь БД – лицо или прикладная программа, которые могут обра-

щаться с командами и/или запросами к БД и получать от нее результаты обращений.

Предметная область – часть реального мира, которая описывается и моделируется с помощью БД.

Прикладная программа – программа, работающая с БД, и написанная на языке программирования, встроенном в СУБД.

Реквизит – характеристика свойства объекта.

Форма – приложение БД, используемое главным образом для ввода, просмотра и редактирования данных.

Экономическая информация – совокупность различных сведений экономического характера, используемых для планирования, учета, контроля, анализа и управления народным хозяйством и его звеньями.

12. МОДЕЛИ ДАННЫХ

12.1. Трехуровневая модель организации баз данных

В настоящее время используется трехуровневая модель организации БД, предложенная в 1975 году комитетом по стандартизации ANSI (American National Standards Institute). Согласно ей одна и та же БД имеет три различные уровни описания: *внешний уровень* (представление о БД отдельных пользователей и прикладных программ); *концептуальный уровень* (БД представляется обобщенно – объединяются данные, используемые различными пользователями и прикладными программами); *внутренний уровень* (поддерживает представление БД в памяти компьютера).

Первоначально исследования в области БД были направлены на разработку способов структуризации данных, получивших название моделей данных. **Модель данных** – это совокупность принципов организации базы данных. Модели данных различаются принципами определения, манипулирования и хранения данных в базе. Но наиболее важным отличительным признаком является способ организации связей между данными в базе.

Классическими являются *иерархическая, сетевая и реляционная* модели данных. Кроме того, в последнее время активно используются при разработке БД такие модели, как *постреляционная, объектно-ориентированная, объектно-реляционная и многомерная* модели.

12.2. Иерархическая модель

В *иерархической модели* связи между данными можно представить с помощью дерева. Данные в такой модели расположены на разных иерархических уровнях и называются *сегментами*. Самый высокий сегмент называется *корневым*. Сегменты, расположенные на более низком уровне, называются *сегментами-потомками*, а расположенные на более высоком уровне, – *сегментами-предками*. Каждый сегмент может иметь только одного предка на более высоком уровне и одного или несколько потомков на более низком уровне. Доступ к

определенному сегменту осуществляется по цепочке, от сегмента-предка к сегменту-потомку, начиная слева. *Достоинством* иерархической модели является эффективное использование памяти компьютера при хранении данных, а *недостатком* – ее громоздкость для обработки данных со сложными логическими связями.

12.3. Сетевая модель

Сетевая модель является развитием иерархической модели. В ней потомок может иметь любое количество предков. Сегменты, которые называются в сетевой модели *наборами записей*, связываются между собой не только по принципу «сверху вниз», но и «по горизонтали» с помощью *наборов связей*. К *достоинствам* сетевой модели данных относятся возможность образования произвольных связей и быстрый доступ к данным. *Недостатками* являются сложность ее понимания для обычного пользователя и большие объемы памяти компьютера на хранение данных.

12.4. Реляционная модель

Реляционная модель данных (РМД) была предложена американским математиком Э. Коддом в 1970 году. В настоящее время это самая распространенная модель данных. В основе РМД лежит понятие *отношения* (relation - англ.). Отношение отображает некоторый объект. Объект характеризуется набором атрибутов D_1, D_2, \dots, D_n , а каждый атрибут – набором допустимых значений, называемым *доменом*. Пусть $D_1 = \{x_1, x_2, \dots, x_k\}$, $D_2 = \{y_1, y_2, \dots, y_l\}$, . . . $D_n = \{z_1, z_2, \dots, z_m\}$. Список имен атрибутов (D_1, D_2, \dots, D_n) называется *схемой отношения*, а количество атрибутов в отношении – *степенью отношения*.

Отношение определяется как *подмножество R декартова произведения* $D_1 \times D_2 \times \dots \times D_n$. Декартово произведение – это набор всевозможных сочетаний из n значений, где каждое значение берется из своего домена. Пусть D_1 содержит номера трех заказов $\{1021, 1022, 1023\}$, D_2 – коды двух клиентов $\{AA, AC\}$, D_3 – веса трех заказов $\{100, 300, 120\}$. В этом случае отношение R есть

декартово произведение $D_1 \times D_2 \times D_3$, которое представляет набор из 18 троек значений, где первое значение – это один из номеров заказов, второе – это один из кодов клиентов, а третье – один из весов заказа.

Создатель РМД Э. Кодд использовал термин «отношение» как синоним слова «таблица». *Столбцы* этой таблицы соответствуют *атрибутам*, а *строки* называются *кортежами*. Количество кортежей в отношении называется *мощностью отношения*.

РМД основана на математическом понятии отношения и представлении отношений в форме таблиц. Таблица в РМД (реляционная таблица) обладает следующими свойствами:

- 1) каждое значение атрибута, содержащееся на пересечении строки и столбца должно быть *атомарным*, т.е. не расчленяться на несколько значений;
- 2) значения в столбце должны быть однородными;
- 3) каждая строка уникальна;
- 4) каждый столбец имеет уникальное имя;
- 5) последовательность строк и столбцов в таблице не существенна;

В таблице реляционной БД *столбцы* называют *полями*, а *строки* – *записями*. Одно или несколько полей, значения которых в каждой записи таблицы однозначно ее идентифицируют, называют *ключевым полем*. В таблице может быть несколько ключевых полей.

В реляционной БД между таблицами устанавливаются *связи* посредством связи ключевых полей, содержащих общую информацию для обеих таблиц. Пусть таблица R_1 связывается с таблицей R_2 . Тогда таблица R_1 именуется *главной*, а таблица R_2 – *подчиненной*. Ключевое поле главной таблицы называется *первичным ключом*, а подчиненной – *внешним ключом*. Одна запись главной таблицы может быть связана с одной или несколькими записями подчиненной таблицы. При этом значения первичного ключа уникальны, а внешнего – могут повторяться.

В общем случае РМД представляет *множество взаимосвязанных таблиц*. Графическое изображение связи между таблицами называется *схемой данных*.

В РМД должны выполняться условия целостности данных. Это условие «целостности таблиц» и условие «ссылочной целостности». Условие «целостности таблиц» накладывает ограничения на значения первичного ключа – значения первичного ключа таблицы должны быть *уникальными и непустыми*. Условие «ссылочной целостности» предполагает, что *каждое значение внешнего ключа должно совпадать с одним из значений первичного ключа*.

В процессе обработки реляционных таблиц СУБД выполняет операции реляционной алгебры, такие как объединение, пересечение, декартово произведение, разность, проекция, выборка, соединение, деление.

Главные *достоинства* РМД заключаются в том, что она проста для понимания, наглядна и имеет строгое математическое обоснование. *Недостатки* проявляются в следующем:

- РМД не допускает представления объектов со сложной структурой, так как в ее рамках возможно моделирование лишь с помощью двумерных таблиц;
- данные об объектах содержатся, как правило, во многих таблицах, и извлечение информации о них требует выполнения многих операций соединения с помощью первичных и внешних ключей, что значительно замедляет обработку данных.

12.5. Постреляционная модель

Постреляционная модель данных в общем случае представляет собой расширенную реляционную модель, снимающую ограничение неделимости значений полей, т.е. допускаются многозначные поля, значения которых состоят из подзначений. *Достоинствами* постреляционной модели данных являются:

- возможность представления связанных реляционных таблиц одной постреляционной таблицей, что обеспечивает высокую наглядность представления данных и повышение эффективности их обработки;
- отсутствие ограничений на длину полей и их количество в записях таблицы.

Недостатком является сложность в обеспечении целостности данных.

12.6. Объектно-ориентированная и объектно-реляционная модели

Объектно-ориентированная и объектно-реляционная модели используются для преодоления ограниченных возможностей реляционной модели по хранению и обработке сложных объектов, как, например, документ, звук, видео и др.

Объектно-ориентированная модель данных (ООМД) представляет структуру, которую можно изобразить графически в виде дерева, узлами которого являются объекты. Каждый объект характеризуется уникальным *идентификатором, состоянием и поведением*. Состояние объекта определяется множеством значений его атрибутов. Поведение объекта описывают *методы*, называемые процедурами. То есть, составной частью описания объекта являются процедуры, способные производить действия над атрибутами объекта в случае наступления тех или иных событий.

Объекты могут объединяться в *классы*. Экземпляры одного класса отличаются лишь значениями своих свойств. Методы устанавливаются при определении класса. Для выполнения действий над объектами применяются объектно-ориентированные механизмы – наследование, инкапсуляция, полиморфизм. *Наследование* – на основе существующего класса можно образовать новый класс объектов, который будет наследовать свойства родительского класса. *Инкапсуляция* – доступ к данным осуществляется только лишь в соответствии с правилами поведения объекта, описываемыми методами. *Полиморфизм* – способность объектов по-разному реагировать на одно и то же событие в окружающем мире. Полиморфизм используется для унификации обработки разнородных объектов.

Основным *достоинством* ООМД является способность отображать информацию о сложных объектах с исчерпывающим описанием взаимосвязей между ними и их динамического поведения. Эта модель обычно применяется для сложных предметных областей, при моделировании которых не хватает функциональности реляционной модели. *Недостатком* является сложность понятийного аппарата, что усложняет ее применение и отрицательно сказыва-

ется на накоплении опыта создания и эксплуатации объектно-ориентированных БД.

Объектно-реляционная модель данных (ООРМД) является гибридной моделью, сочетающей возможности реляционной модели с объектными свойствами данных. Ее отличительная особенность состоит в том, что она основана на стратегии реляционной модели.

12.7. Многомерная модель

Многомерность модели означает *многомерное логическое представление структуры информации*, а не многомерность визуализации данных. Многомерная модель предназначена для аналитической обработки информации. Ее принципы организации данных – это агрегируемость, историчность, прогнозируемость данных. *Агрегируемость* данных означает возможность их рассмотрения с различным уровнем обобщения. *Историчность* – статичность (неизменяемость) данных и их взаимосвязей, а также привязка их к временным точкам. *Прогнозируемость* – задание функций прогнозирования и применение их к различным интервалам времени.

Основные понятия многомерной модели – *измерение и ячейка*. **Измерение** – это множество однотипных данных, образующих одну из граней многомерного гиперкуба. Используются измерения временные (дни, месяцы, кварталы и годы) и географические (города, районы, регионы и страны). **Ячейка** – это поле, значение которого однозначно определяется фиксированным набором измерений.

В многомерной модели данных используется два варианта организации данных – гиперкубическая и поликубическая. В *гиперкубической* все кубы определяются одним и тем же набором измерений (максимально возможным). В *поликубической* определяются несколько гиперкубов с различной размерностью и различными измерениями в качестве граней.

Для извлечения данных из базы, организованной по многомерной модели, применяется ряд специальных операций: срез, вращение, агрегация и дета-

лизация. *Срез* представляет собой данные, полученные в результате фиксации одного или нескольких измерений. Операция *вращение* – процедура изменения порядка следования измерений. Операции *агрегации* и *детализации* означают соответственно переход к более или менее детальному представлению информации из гиперкуба.

Достоинством многомерной модели является удобство и эффективность аналитической обработки больших объемов данных, связанных с временными интервалами; *недостатком* – громоздкость для простейших задач оперативной обработки информации.

Необходимо отметить, что исследования в области моделей данных продолжаются.

Глоссарий

Внешний ключ – набор полей в подчиненной таблице, по которому она связывается с основной таблицей.

Домен – набор допустимых значений атрибута.

Иерархическая модель – модель, у которой данные, называемые сегментами, расположены на разных иерархических уровнях и каждый сегмент может иметь только одного предка на более высоком уровне и одного или несколько потомков на более низком уровне.

Ключевое поле – одно или несколько полей, значения которых в каждой записи таблицы однозначно ее идентифицируют.

Кортеж – строка таблицы, являющейся отношением.

Многомерная модель – модель с многомерным логическим представлением структуры информации.

Модель данных – совокупность принципов организации данных в базе данных.

Мощность отношения – количество кортежей в отношении.

Объектно-ориентированная модель данных – структура данных, имеющая графическое изображение в виде дерева, узлами которого являются объекты.

Объектно-реляционная модель данных – реляционная модель данных с объ-

ектными свойствами данных.

Отношение – именованная структура данных в реляционной БД, представляющая собой подмножество декартова произведения заданных доменов.

Первичный ключ – уникальный идентификатор записей таблицы.

Постреляционная модель данных – реляционная модель данных, у которой снято ограничение неделимости значений полей.

Реляционная модель данных – модель данных, основанная на математическом понятии отношения и представлении отношений в форме таблиц.

Сетевая модель данных – модель данных, у которой потомок может иметь любое количество предков.

Степень отношения – количество атрибутов в отношении.

Схема данных – графическое изображение связи таблиц в реляционной базе данных.

Схема отношения – список имен атрибутов отношения.

Условие ссылочной целостности – каждое значение внешнего ключа должно совпадать с одним из значений первичного ключа.

Условие целостности таблиц – значения первичного ключа таблицы должны быть непустыми.

13. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

13.1. Требования, предъявляемые к БД

Проектирование БД – это процесс создания проекта БД, предназначенной для поддержки функционирования экономического объекта и способствующей достижению его целей. При проектировании БД необходимо учитывать следующие требования:

- 1) целостность БД (полнота и непротиворечивость данных);
- 2) многократное использование данных;
- 3) быстрый поиск и получение информации по запросам пользователей;
- 4) простота обновления данных;
- 5) уменьшение излишней избыточности данных;
- 6) защита данных от несанкционированного доступа, от искажения и уничтожения.

13.2. Этапы жизненного цикла БД

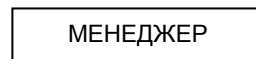
Жизненный цикл БД (ЖЦБД) – это процесс проектирования, реализации и поддержки БД. Он состоит из следующих семи этапов:

- 1) предварительное планирование БД;
- 2) проверка осуществимости;
- 3) определение требований к БД;
- 4) концептуальное проектирование (разработка концептуальной модели БД);
- 5) логическое проектирование (разработка логической модели БД);
- 6) физическое проектирование (разработка БД в среде некоторой СУБД);
- 7) оценка работы и поддержка БД.

13.3. Модель «сущность-связь»

Для моделирования данных предметной области на этапе *концептуального* проектирования БД широко используется модель «сущность-связь» (ER-модель). Эта модель позволяет графически представить объекты и связи между ними.

Сущность – это некоторый объект реального мира, который может существовать независимо. **Атрибут** – это свойство сущности. Сущность имеет экземпляры, отличающиеся друг от друга значениями атрибутов и допускающие однозначную идентификацию. Атрибут, который уникальным образом идентифицирует экземпляры сущности, называется *ключевым*. Может быть *составной* ключ, представляющий комбинацию нескольких атрибутов. Сущность изображается прямоугольником, например:



Связь представляет взаимодействие между сущностями. Она характеризуется *мощностью*, которая показывает, сколько сущностей участвует в связи. Связь между двумя сущностями называется *бинарной*, а связь между более чем с двумя сущностями – *n-арной*.

Пусть требуется разработать ER-модель предметной области БАНК, описанной ниже:

Банк имеет филиалы, которые управляются менеджерами. Филиалы обрабатывают счета клиентов.

В данной предметной области можно выделить сущности МЕНЕДЖЕР, ФИЛИАЛ, КЛИЕНТ, СЧЕТ и следующие связи между ними

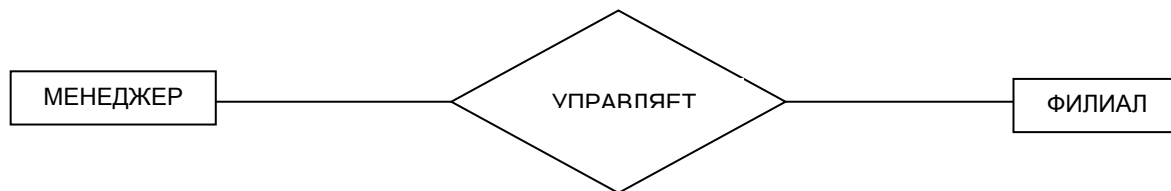
1. МЕНЕДЖЕР – УПРАВЛЯЕТ – ФИЛИАЛ
2. ФИЛИАЛ – ОБРАБАТЫВАЕТ – СЧЕТ
3. КЛИЕНТ – ИМЕЕТ – СЧЕТ

Связь изображается ромбом, например:



Существует три типа связи. Пусть А и В некоторые сущности.

Если каждый экземпляр сущности **A** может быть связан не более чем с одним экземпляром сущности **B**, то связь между сущностями **A** и **B** имеет тип "один-к-одному" (1:1). Так в нашем примере связь **1** типа 1:1, и ER-диаграмма для этой связи имеет вид:



Если каждый экземпляр сущности **A** может быть связан более чем с одним экземпляром сущности **B**, а каждый экземпляр сущности **B** может быть связан не более чем с одним экземпляром сущности **A**, то связь между сущностями **A** и **B** имеет тип "один-ко-многим" (1:M).

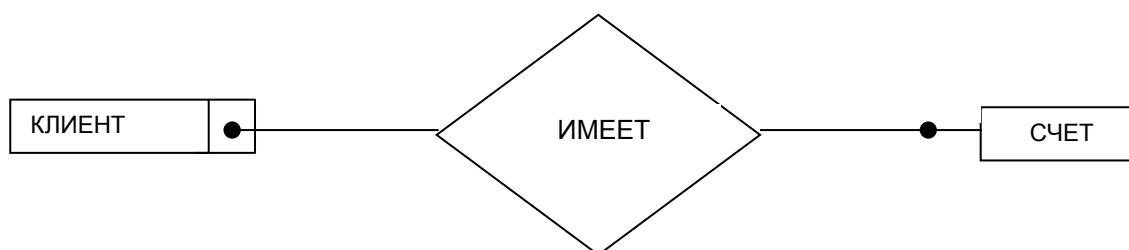
Если каждый экземпляр сущности **A** может быть связан с несколькими экземплярами сущности **B**, и каждый экземпляр сущности **B** может быть связан с несколькими экземплярами сущности **A**, то связь между сущностями **A** и **B** имеет тип "многие-ко-многим" (M:N).

Рассмотрим понятие *класса принадлежности (КП) сущности*.

Если каждый экземпляр сущности **A** связан с экземпляром сущности **B**, то КП сущности **A** является *обязательным*, что отмечается на ER-диаграмме кружочком, помещенным в прямоугольник, смежный с прямоугольником сущности **A**.

Если не каждый экземпляр сущности **A** связан с экземпляром сущности **B**, то КП сущности **A** является *необязательным*, что отмечается на ER-диаграмме кружочком на линии связи возле прямоугольника сущности **A**.

На нижеприведенной ER-диаграмме КП сущности клиент обязательный, а сущности счет – необязательный.



Предположим, что в рассматриваемой предметной области **БАНК** КП

всех 4-х сущностей является обязательным. Тогда ER-модель предметной области **БАНК** будет иметь вид, как на рис. 13.1.

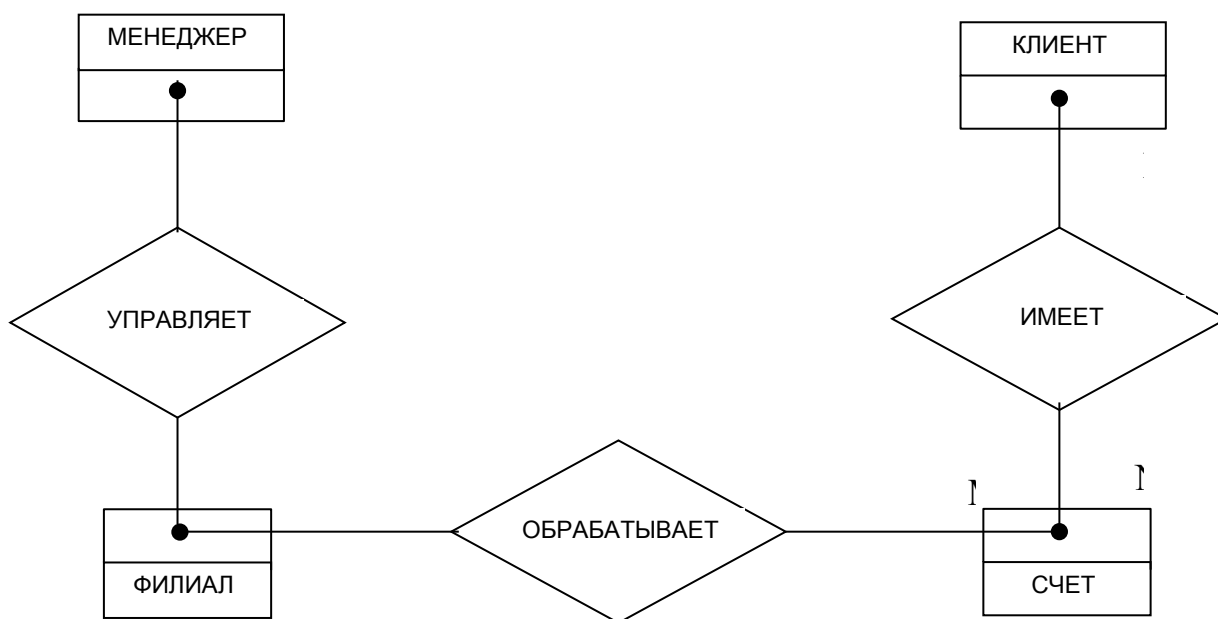


Рис. 13.1. ER-модель предметной области **БАНК**

Затем каждая из 4-х сущностей приведенной ER-модели описывается своим

набором атрибутов (см. рис. 13.2).

МЕНЕДЖЕР
Номер менеджера (НМ)
Стаж работы (СТАЖ)
Специальность (СПЕЦ)

ФИЛИАЛ
Номер филиала (НФ)
Адрес филиала (АДР_Ф)

СЧЕТ
Номер счета (НС)
Тип счета (ТИП)
Остаток на счете (ОСТ)

КЛИЕНТ
Номер клиента (НК)
Ф.И.О. клиента (ФИО_К)
Социальное положение (СОЦ)
Адрес клиента (АДР_К)

Рис. 13.2. Наборы атрибутов сущностей предметной области **БАНК**

с выделенными первичными ключами

ER-модель в совокупности с наборами атрибутов сущностей представляет собой *концептуальную модель* предметной области или ее *концептуальную*

схему БД.

13.4. Преобразование ER-модели в реляционную модель

Пусть для создания логической модели БД выбирается реляционная модель данных. Тогда для каждой сущности создается таблица, в которой каждому атрибуту соответствует столбец таблицы.

Существует шесть правил генерации таблиц из ER-диаграмм. Они опираются на два основных фактора – *тип связи* и *КП сущности*.

Для связи типа 1:1 существуют три правила формирования предварительных таблиц из ER-диаграмм.

Правило 1

Если связь типа 1:1 и КП обеих сущностей является обязательным, то создается только одна таблица. Первичным ключом (ПК) этой таблицы может быть ПК любой из двух сущностей.

Правило 2

Если связь типа 1:1 и КП одной сущности является обязательным, а другой – необязательным, то создается две таблицы (по одной для каждой сущности). ПК сущности должен быть ПК соответствующей таблицы. ПК сущности, для которой КП является необязательным, добавляется как атрибут в таблицу для сущности с обязательным КП.

Правило 3

Если связь типа 1:1 и КП обеих сущностей является необязательным, то создаются три таблицы – по одной для каждой сущности и одна для связи. ПК сущности должен быть ПК соответствующей таблицы. Таблица для связи должна иметь среди своих атрибутов ключи обеих сущностей.

Для связи типа 1:М существует только два правила. Выбор одного из них зависит от КП сущности на стороне М (КП сущности на стороне 1 не влияет на выбор).

Правило 4

Если связь типа 1:М и КП сущности на стороне М является обязательным, то необходимо построить две таблицы (по одной для каждой сущности). ПК сущности должен быть ПК соответствующей таблицы. ПК сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне М.

Правило 5

Если связь типа **1:M** и *КП* сущности на стороне *M* является *необязательным*, то необходимо построить три таблицы – по одной для каждой сущности и одну для связи. ПК сущности должен быть ПК соответствующей таблицы. Таблица для связи должна иметь среди своих атрибутов ключи обеих сущностей.

Для связи типа *M:N* КП сущности не имеет значения и существует одно правило

Правило 6

Если *связь типа M:N*, то необходимо построить три таблицы – по одной для каждой сущности и одну для связи. ПК сущности должен быть ПК ключом соответствующей таблицы. Таблица для связи должна иметь среди своих атрибутов ключи обеих сущностей.

13.5. Нормализация таблиц

Нормализация таблиц – это процесс, позволяющий минимизировать избыточность данных. Он сводится к приведению таблиц как минимум к третьей нормальной форме (3НФ).

Первая нормальная форма (1НФ) определяется следующим образом:

Таблица находится в 1НФ, если все ее поля содержат только неделимые значения.

На практике, если в клетках столбца содержится несколько значений, то каждое из них представляется отдельной записью.

Вторая нормальная форма (2НФ) базируется на понятии частичной и полной функциональной зависимости и определяется следующим образом:

Таблица находится в 2НФ, если она удовлетворяет требованиям 1НФ и неключевые поля функционально полно зависят от первичного ключа.

На практике, если неключевые поля находятся в частичной функциональной зависимости, то они удаляются из таблицы и помещаются в новую таблицу совместно с подмножеством первичного ключа, от которого они зависят.

Третья нормальная форма (3НФ) определяется следующим образом:

Таблица находится в 3НФ, если она удовлетворяет требованиям 2НФ и не содержит транзитивных зависимостей

Транзитивной зависимостью называется функциональная зависимость

между неключевыми атрибутами. На практике, если неключевые поля находятся в транзитивной зависимости, то они помещаются в отдельные таблицы, вместе с ключевыми полями.

13.6. Этапы проектирования базы данных и их процедуры

Проектирование БД состоит из трех этапов: концептуального, логического и физического проектирования.

Процедуры *концептуального* проектирования следующие.

1. Определение сущностей и их документирование.
2. Определение связей между сущностями и их документирование.
3. Создание ER-модели предметной области.
4. Определение атрибутов сущностей и их документирование.
5. Определение доменов атрибутов и их документирование.
6. Определение первичных ключей сущностей и их документирование.
7. Обсуждение концептуальной модели данных с конечными пользователями.

Процедуры *логического* проектирования следующие.

1. Выбор модели данных.
2. Определение набора таблиц исходя из ER-модели и их документирование.
3. Нормализация таблиц.
4. Проверка логической модели данных на предмет возможности выполнения всех операций, предусмотренных пользователями.
5. Определение требований поддержки целостности данных и их документирование.
6. Создание окончательного варианта логической модели данных и обсуждение его с пользователями.

Процедуры *физического* проектирования следующие.

1. Выбор СУБД
2. Проектирование таблиц БД средствами выбранной СУБД.
3. Реализация бизнес-правил в среде выбранной СУБД.
4. Проектирование физической организации БД.

5. Разработка стратегии защиты БД.
6. Организация мониторинга БД и ее настройка.

13.7. Семантическая объектная модель

На этапе концептуального проектирования БД может использоваться подход, отличный от ER-модели, – семантическая объектная модель (СОМД). В ее основе лежит понятие семантического (смыслового) объекта. **Семантический объект** – это представление некоторой вещи, идентифицируемой в рабочей среде пользователя. Подобно сущностям он имеет набор атрибутов, являющийся достаточным описанием объекта. Выделяют три типа атрибутов:

- 1) *простые* атрибуты состоят из одного элемента, например, Код клиента;
- 2) *групповые* – совокупности нескольких атрибутов, например, Адрес (Улица, Город, Республика, Индекс);
- 3) *семантические объектные* атрибуты – это атрибуты, которые устанавливают связь между двумя семантическими объектами.

Создаются *объектные диаграммы*. На них объекты изображаются как вертикально ориентированные прямоугольники. Пусть необходимо представить объект Кафедра. Тогда Название кафедры, Телефон – его простые атрибуты, Адрес – групповой, УНИВЕРСИТЕТ, ПРЕПОДАВАТЕЛЬ, СТУДЕНТ – семантические объектные атрибуты, они связаны с объектом КАФЕДРА и логически содержатся в нем.

Один или несколько объектных атрибутов, с помощью которых пользователи идентифицируют экземпляры объекта – это *объектный идентификатор*. Он может быть простым и групповым и обозначается с помощью букв ID перед ним. Если идентификатор уникальный, то эти буквы подчеркивают.

Каждый атрибут семантического объекта имеет максимальное и минимальное кардинальные числа. *Минимальное кардинальное число* показывает количество значений атрибута, которые должны существовать, чтобы объект был допустимым. Обычно оно равно **0** или **1**. Если 0, то атрибут не обязан иметь значение; если 1 – обязан иметь значение. *Максимальное кардинальное число*

показывает максимальное количество значений атрибута, которое может иметь объект. Обычно оно равно 1 или N. Если 1, то атрибут может иметь не более одного значения, если N – то много значений.

Кардинальность изображается в виде нижнего индекса атрибута в формате N.M, где N – минимальное, а M – максимальное кардинальное число.

С учетом сказанного объектная диаграмма для объекта КАФЕДРА будет иметь вид, как на рис. 13.1.

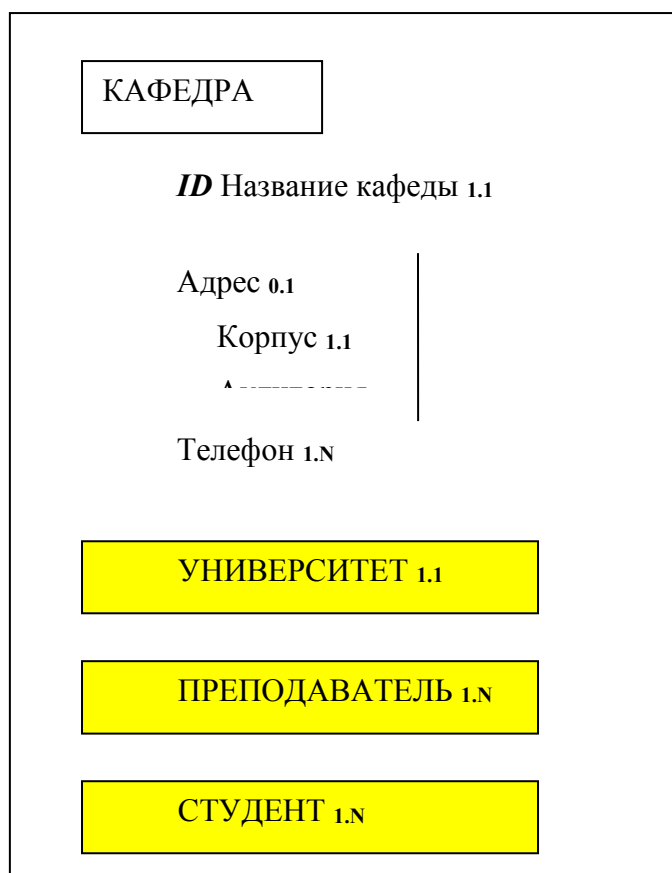


Рис. 13.1. Пример объектной диаграммы

В СОМД объектные атрибуты должны быть *парными*. Если один объект содержит в себе другой, то этот другой должен содержать в себе первый. Так, объект УНИВЕРСИТЕТ должен содержать объект КАФЕДРА.

Для каждого объекта предметной области составляется объектная диаграмма – в результате получается СОМД.

Отличия СОМД от модели «сущность-связь»:

- в качестве базовых элементов рассматриваются не сущности, а семантические объекты;
- данная модель содержит больше информации о значениях данных, чем модель «сущность-связь».

13.8. CASE-средства для моделирования данных

ER-модели получили широкое распространение в CASE-средствах – программных средствах для автоматизированного проектирования реляционных БД. Широко распространены CASE-системы: **ERwin**, **Design/IDEF**, **Power Designer**. Их графические средства моделирования предметной области дают возможность наглядно изучать концептуальную модель данных и перестраивать ее соответственно поставленным целям и имеющимся ограничениям. Охарактеризуем CASE-средство **Erwin**.

ERwin позволяет строить *логическую* модель БД в различных предметных областях и автоматически генерировать *физическую* модели БД – создавать структуры данных в среде многих настольных и серверных СУБД.

ERwin имеет развитые средства создания различных представлений ER-диаграмм:

- *диаграмм уровня сущностей*. На них приводятся сущности и их связи. Они удобны для обзора большой модели;
- *диаграмм уровня определений*. На них даются определения сущностей. Они служат для презентации диаграмм другим людям;
- *диаграмм уровня атрибутов*. На них прямоугольник-сущность делится линией на 2 части – в верхней отображаются атрибуты (колонки) первичного ключа, а в нижней – остальные атрибуты (колонки). Они используются при создании логической и физической моделей БД;
- *диаграмм уровня первичных ключей*. На них внутри прямоугольников-сущностей показываются только атрибуты, составляющие первичный ключ;
- *диаграмм уровня иконок*. На них сущностям ставятся в соответствие иконки. Они служат для презентационных целей.

ERwin позволяет:

- редактировать и оформлять модель БД;
- получать подробные отчеты по модели БД;
- осуществлять прямое и обратное моделирование БД. *Прямое моделирование* – это описание схемы БД в графическом виде, а затем получение готовой БД. *Обратное моделирование* – это восстановление схемы БД по существующей БД, что необходимо при переносе БД из среды одного сервера БД в среду другого;
- моделирование структуры БД по стандарту IE и стандарту IDEF1X. Наиболее распространен стандарт *IDEF1X*– это регламентация разработки структуры БД с нуля (прямого моделирования БД);
- коллективную разработку модели БД.

Перечисленные инструменты **ERwin** помогают свести рутинный труд разработчика к минимуму, снизить потери времени при согласовании моделей БД со специалистами предметной области, облегчить поддержку созданных информационных систем в будущем.

Глоссарий

ER-диаграмма связи – графическое представление связи между двумя сущностями.

Бинарная связь – связь между двумя сущностями.

Вторая нормальная форма (2НФ) – таблица находится в 2НФ, если она удовлетворяет требованиям 1НФ и неключевые поля функционально полностью зависят от первичного ключа.

Жизненный цикл базы данных – процесс проектирования, реализации и поддержки базы данных.

Класс принадлежности сущности необязательный – не каждый экземпляр сущности А связан с экземпляром сущности В.

Класс принадлежности сущности обязательный – каждый экземпляр сущ-

ности А связан с экземпляром сущности В.

Ключ сущности – атрибут, который уникальным образом идентифицирует экземпляры сущности.

Концептуальное проектирование – создание концептуальной модели данных исходя из представлений пользователей о предметной области.

Логическое проектирование – преобразование концептуальной модели на основе выбранной модели данных в логическую модель, не зависимую от особенностей используемой в дальнейшем СУБД для физической реализации базы данных

Модель «сущность–связь» (ER-модель) – модель, у которой структуры данных предметной области представляются с помощью графических средств – диаграмм «сущность–связь» (ER-диаграмм).

Мощность связи – количество сущностей, участвующих в связи.

Нормализация таблиц – процесс, позволяющий минимизировать избыточность данных.

Первая нормальная форма (1НФ) – таблица находится в 1НФ, если все ее поля содержат только неделимые значения.

Проектирование базы данных – процесс создания проекта БД, предназначенной для поддержки функционирования экономического объекта и способствующей достижению его целей.

Связь между сущностями – взаимодействие между сущностями

Связь типа «многие-ко-многим» (M:N) – каждый экземпляр сущности А может быть связан с несколькими экземплярами сущности В и каждый экземпляр сущности В может быть связан с несколькими экземплярами сущности А.

Связь типа «один-к-одному» (1:1) – каждый экземпляр сущности А может быть связан не более чем с одним экземпляром сущности В.

Связь типа «один-ко-многим» (1:M) – каждый экземпляр сущности А может быть связан более чем с одним экземпляром сущности В, а каждый экземпляр сущности В может быть связан не более чем с одним экземпляром сущности

А.

Семантический объект – представление некоторой вещи, идентифицируемой в рабочей среде пользователя.

Составной ключ сущности – комбинация нескольких атрибутов, однозначно определяющая экземпляры сущности.

Сущность – некоторый объект реального мира, который может существовать независимо

Третья нормальная форма (3НФ) – таблица находится в 3НФ, если она удовлетворяет требованиям 2НФ и не содержит транзитивных зависимостей.

Физическое проектирование – описание конкретной реализации базы данных, размещаемой во внешней памяти компьютера.

Экземпляр сущности – конкретный объект, характеризующийся набором значений атрибутов сущности.

14. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫМИ

14.1. Понятие СУБД

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

Языковые средства – это язык описания данных (ЯОД), язык манипулирования данными (ЯМД), язык запросов, язык программирования. ЯОД – средства для описания структуры данных, их типов, ограничений для данных. ЯМД позволяет вставлять, обновлять, удалять и извлекать данные. Язык запросов, как например язык SQL, позволяет выполнение всех операций над данными БД. Язык программирования, встроенный в СУБД, дает возможность создавать программы для глубокой аналитической обработки данных базы.

Программные средства СУБД обеспечивают работу с физической БД (поиск, изменение, обработка, извлечение данных и др.) и выполнение всех ее функций.

14.2. Архитектура СУБД

Архитектура СУБД – это совокупность функциональных компонентов системы и их взаимосвязей. СУБД включает три компонента:

- *подсистему средств проектирования БД и ее приложений.* Это набор инструментов, упрощающих проектирование и реализацию БД и ее приложений;
- *подсистему обработки компонентов приложений.* Она обеспечивает работу приложений БД и включает: процессор запросов, процессор форм, генератор отчетов; средства обработки, реализованные на языках программирования;

ядро СУБД, которое является посредником между данными подсистемами и БД. Оно выполняет чтение и запись данных при обращении к внешним устройствам, участвует в блокировке, резервном копировании и восстанов-

лении БД, управлении транзакциями. **Транзакция** – это логическая единица работы, включающая несколько команд вставки, удаления или модификации, которая переводит БД из одного завершенного состояния в другое завершенное состояние.

14.3. Классификация СУБД

Классифицировать СУБД можно по:

1) *по степени универсальности*: СУБД общего назначения (они не ориентированы на какую-либо конкретную сферу применения и обладают средствами настройки на работу с определенной БД) и СУБД специального назначения, например, **IMBASE** – СУБД для автоматизации проектных и конструкторских работ;

2) *по типу поддерживаемой модели данных*:

- иерархические. Первая иерархическая СУБД – **IMS** компании IBM (1968г.);
- сетевые. Первая сетевая СУБД – **IDS** компании General Electric;
- реляционные. Первые коммерческие реляционные СУБД – от компаний IBM, Oracle Corporation и др.;
- постреляционные, например, **uniVers**, **Bubba**, **Dasdb**;
- объектно-ориентированные, например, **ORION**, **IRIS**, **Vbase**, **PDM**;
- объектно-реляционные, например, **Informix Universal Server** (Informix Software), **DB2 Universal Database** (IBM);
- многомерные, например, **Oracle Express Server** (Oracle), **Cache** (InterSystems);

3) *по принципу обработки запросов к БД*: настольные, серверные, распределенные.

14.4. Возможности, предоставляемые СУБД пользователям

СУБД предоставляют пользователям следующие основные возможности:

- создание БД и обновление хранящихся в ней данных;
- быстрое извлечение из БД необходимых данных по запросам пользователей;

- выполнение вычислений над данными;
- создание форм для удобства работы с данными базы;
- создание отчетов на основе информации БД для ее анализа пользователями;
- разработка приложений, работающих с БД;
- экспорт/импорт данных в БД из других приложений;
- публикация БД в Internet;
- управление БД: поддержка целостности БД, защита данных, восстановление БД в случае ее повреждения.

14.5. Производительность СУБД

Наиболее важные показатели производительности СУБД:

- скорость выполнения операций корректировки данных;
- время выполнения запросов (время отклика);
- время генерации отчета;
- максимальное число одновременных обращений к БД в многопользовательском режиме.

Для сравнительной оценки производительности коммерческих СУБД имеется ряд эталонных тестов, разработанных международной организацией ТРС.

14.6. Режимы работы пользователя в СУБД

Пользователь может работать с БД в СУБД в режимах:

- *через меню системы* – команды выбираются из различных меню и их параметры уточняются в диалоговых окнах;
- *командный режим* – с клавиатуры вводятся команды языка запросов SQL;
- *программный режим* – организация доступа к данным и управление ими производится из прикладных программ.

Работа с локальной БД осуществляется в *однопользовательском* режиме, с централизованной БД – в *многопользовательском*.

14.7. Функции СУБД

СУБД выполняет следующие функции.

1. Управление: данными во внешней памяти, буферами оперативной памяти, параллельным доступом, транзакциями.
2. Поддержка: языков БД, обмена данными (экспорт/импорт), целостности данных, независимости прикладных программ от данных.
3. Ведение системного каталога (словаря данных).
4. Контроль доступа к данным.

14.8. Направления развития СУБД

Направления развития СУБД можно сформулировать следующим образом.

1. Расширение множества типов обрабатываемых данных. Современные СУБД позволяют хранение и обработку не только данных традиционного типа (чисел, текстов, дат), но и других типов: логических данных, графических изображений, документов, звука, видео, гиперссылок и др.
2. Комбинирование технологий WWW и технологий БД, позволяющее опубликовывать данные базы на Web-страницах и работать с ними на удаленном от БД расстоянии с помощью браузеров.
3. Превращение СУБД в системы управления базами знаний (о базах знаний речь пойдет в теме 19).

Глоссарий

Архитектура СУБД – представление о совокупности функциональных компонентов СУБД и их взаимосвязях.

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

Транзакция – логическая единица работы, включающая несколько команд вставки, удаления или модификации, которая переводит БД из одного завершеного состояния в другое завершеного состояние.

Язык запросов – ориентированный на конечного пользователя язык доступа к БД.

Язык манипулирования данными – язык, позволяющий выполнять операции над базой данных, такие как вставка, обновление, удаление и извлечение данных из базы.

Язык описания данных – язык, служащий для описания структуры данных, их типа, а также для задания ограничений для данных базы.

15. ОБЩАЯ ХАРАКТЕРИСТИКА СУБД MICROSOFT ACCESS 2010

15.1 Характеристики Microsoft Access 2010

СУБД Microsoft Access 2010 (в дальнейшем Access) – программная система для работы с реляционными БД в информационных системах. Это приложение для создания и ведения баз данных. Его понятный и доступный графический пользовательский интерфейс, множество встроенных шаблонов БД, современные инструменты и функции обработки данных позволяют даже непрофессионалу работать с большими объемами данных, отслеживать ключевую информацию, упрощать управление и анализ данных.

Access является компонентом офисного пакета Microsoft Office 2010. Содержит инструментальные средства для создания и работы с локальными и многопользовательскими БД. Предоставляет пользователям доступ к данным в глобальных сетях. Позволяет создавать web-базы данных и публиковать их на сайтах SharePoint, посетители которых могут работать с БД через браузер на основе разрешений SharePoint, определяющих возможность доступа к объектам базы.

Большие возможности по экспорту/импорту данных в БД. Имеет встроенный язык программирования Visual Basic for Applications (VBA). Содержит средства для администрирования БД (защиты, восстановления, сжатия, преобразования БД к формату БД предыдущих версий: Access 2000, Access 2002-2003, Access 2007). Имеет развитую справочную систему, включающую справочные материалы с web-сайта Office.com.

15.2. Характеристика объектов базы данных

СУБД Access позволяет создавать реляционные БД, объектами которых являются: таблицы, запросы, формы, отчеты, макросы и модули.

Таблица предназначена для хранения информации пользователя в БД. Ее строки – это записи, а столбцы – поля БД. Записи идентифицируются ключом таблицы. Таблицам и их полям присваиваются символьные имена, в которых

запрещены: точка (.), восклицательный знак (!), надстрочный символ (^), прямые скобки (||). Между таблицами возможна установка связей двух типов – 1:1, 1:M. Для этого имеются средства проектирования схем данных.

Запрос может использоваться для просмотра, добавления, изменения или удаления данных из БД, а также для фильтрации данных, выполнения расчетов на основе данных и отображения сводных данных.

Форма используется с целью создания интерфейса пользователя для БД. Формы ускоряют работу с БД и делают ее более приятной и эффективной. Существуют «привязанные» и свободные формы. «Привязанная» форма, в отличие от свободной, напрямую соединена с источником данных (таблицей или запросом) и может использоваться для ввода, изменения или отображения данных базы. Формы могут содержать кнопки, надписи и другие элементы управления, необходимые для работы с БД. Можно построить составную форму, включающую другие формы.

Отчет позволяет выводить на экран или на печать информацию из БД в виде, удобном для ее восприятия и анализа пользователем. В отчете можно группировать и сортировать данные, осуществлять расчеты в строках и проводить итоговые вычисления над группами строк и над всеми строками с использованием статистических функций. Отчет может основываться на таблице или запросе. Он может быть составным (включать другие отчеты). Можно создать отчет с адресом, отформатированным для печати наклеек на почтовые конверты.

СУБД Access предоставляет большие возможности по оформлению отчетов и форм: шрифтовое, фоновое и цветное оформление, обрамление, рисунки, деловая графика, темы, вставка объектов других приложений и др., а также по просмотру и печати отчетов.

Макрос предназначен для автоматизации работы с формами и отчетами без написания программ. Это набор из одной или нескольких макрокоманд, каждая из которых выполняет определенное действие. Макрокоманды могут

иметь аргументы, уточняющие действие, выполняемое ими.

Модуль представляет собой программу для работы с БД, написанную на языке VBA.

Все объекты БД можно хранить в одном файле – *файле БД* с расширением .accdb. Это упрощает их перенос с компьютера на компьютер, облегчает создание связанных объектов, проверку целостности данных.

Некоторые ограничения в Access приведены в табл. 15.1.

Таблица 15.1

Спецификации Access

Атрибут	Максимальное значение
Число объектов в БД	32 768
Число одновременно работающих пользователей с БД	255
Число полей в таблице	255
Число одновременно открытых таблиц	2048
Число таблиц в запросе	32
Число печатных страниц в отчете	65 536
Число макрокоманд в макросе	999

Для создания объектов БД существуют такие инструментальные средства, как Конструкторы и Мастера. *Конструктор* предоставляет пользователю ряд инструментальных средств, с помощью которых можно быстро и просто создавать и модифицировать объекты БД. *Мастер* последовательно задает пользователю вопросы и на основе его ответов строит законченный объект БД или выполняет определенную операцию.

15.3. Пользовательский интерфейс

Меню и панели инструментов, которые использовались в предыдущих версиях Access, заменены в Access 2010 лентой. Лента представляет собой основную часть пользовательского интерфейса и представляет набор вкладок, на

которых расположены группы команд, упорядоченных по выполняемым функциям (рис. 15.1).

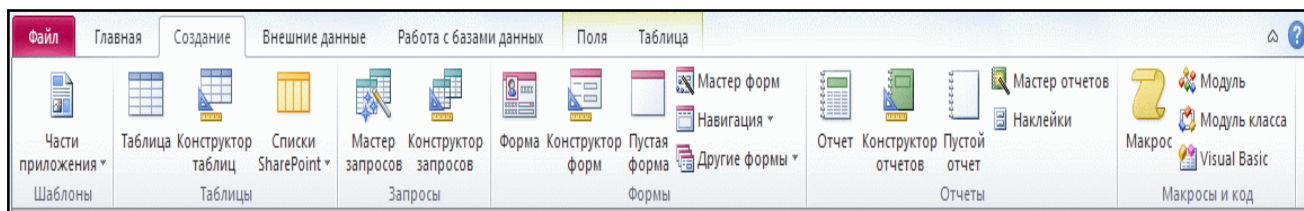


Рис. 15.1. Вид ленты с активной вкладкой Создание

Различают:

- *вкладки команд* – вкладки, на которых команды отображаются и группируются в соответствии со способом их использования, чтобы их можно было легко найти в нужный момент;
- *контекстные вкладки команд* – вкладки, появляющиеся в определенном контексте, т. е. в зависимости от того, над каким объектом ведется работа и какие задачи выполняются. Контекстная вкладка команд содержит команды, которые с наибольшей вероятностью могут потребоваться для работы;
- *коллекции* – элементы управления, которые позволяют просмотреть результат применения стиля или параметра, прежде чем сделать выбор.

В процессе работы с БД в СУБД Access пользователь имеет дело с различными окнами: окном Access, окнами объектов БД, окнами Конструкторов и Мастеров, окнами сообщений и др.

15.4. Настройка рабочей среды

В Access настройку рабочей среды обеспечивает команда **Файл/Параметры**. В частности, возможны настройка:

- *ленты* в соответствии с индивидуальными нуждами пользователя. Например, можно создать настраиваемые вкладки и группы с часто используемыми командами;
- *Панели быстрого доступа*, содержащей набор команд, не зависящих от вкладки, отображаемой в данный момент на ленте. Можно добавлять на панель кнопки, выполняющие команды;

- *параметров пользователей для текущей БД* таких, как отображение окон объектов, включения сочетаний клавиш и параметров автоматического исправления для текущей БД;
- *различных параметров баз данных Access по умолчанию;*
- *Конструкторов объектов БД;*
- *внешнего вида таблиц;*
- *задание языковых параметров и др.*

15.5. Типы данных и выражения

СУБД Access может обрабатывать данные полей следующих типов.

Текстовый. Текст или комбинация текстовых и числовых значений до 255 СИМВОЛОВ.

Поле МЕМО. Длинный текст (до 63999) символов.

Числовой. Числовые данные. Размеры числовых полей – 1, 2, 4, 8, 12 байтов.

Дата/время. Значения даты и времени. Даты от 100 по 9999 год.

Денежный. Денежные значения (максимум 15 знаков в целой части и 4 знака в дробной).

Счетчик. Уникальные последовательные с шагом 1 номера, автоматически присваиваемые записям при их вставке в таблицу БД.

Логический. Поле этого типа может иметь одно значение из пары значений Да/Нет.

Поле объекта OLE. Объекты других приложений Microsoft, которые могут быть связаны или внедрены в таблицу БД Access.

Гиперссылка. Путь к файлу на жестком диске либо адрес в сети Internet или Интранет.

Вложение. Цифровые изображения и двоичные файлы любого типа.

Вычисляемый. Выражения для расчетов над значениями полей таблицы БД.

Каждому типу поля присущ определенный набор свойств. *Свойства* позволяют управлять сохранением, обработкой и отображением данных по-

ля. Некоторые свойства полей:

Размер поля. Определяет максимальную длину текстового или числового поля. Так, для числового поля с целыми значениями – размеры, называемые:

- байт – целые числа от 0 до 255 (поле занимает 1 байт);
- целое – целые числа от -32 768 до 32 767 (2 байта);
- длинное целое – целые числа от -2 147 483 648 до 2 147 483 647 (4 байта).

Для числовых полей с дробными значениями – размеры: одинарное с плавающей точкой, двойное с плавающей точкой, действительное.

Формат поля. Определяет способ отображения текста, чисел, дат и значений времени на экране и на печати.

Число десятичных знаков. Позволяет указывать для чисел количество дробных знаков.

Маска ввода. Задает маску ввода, облегчающую ввод данных в поле.

Подпись. Определяет текст, который выводится в качестве наименований полей в таблицах, запросах, формах, отчетах, СДКД.

Значение по умолчанию. Позволяет указать значение, автоматически вводимое в поле при создании новой записи.

Условие на значение. Определяет множество значений, которые можно вводить в поле таблицы.

Сообщение об ошибке. Позволяет указать текст сообщения, выводимого на экран при вводе недопустимого значения.

Обязательное поле. Указывает, требует ли поле обязательного ввода значения.

Пустые строки. Определяет, допускается ли ввод в текстовое поле пустых строк (" ").

Индексированное поле. Служит для создания индекса.

Сжатие Юникод. Определяет необходимость хранения в поле текста, если его размер меньше 4096 знаков и др.

Выражения используются при определении условий в запросах и макросах, при вычислениях в формах, отчетах и др. **Выражение** – это комбинация

некоторых или всех компонентов: функций, идентификаторов, констант и операторов. Результатом вычисления каждого выражения является одно значение. *Функции* – это небольшие программы, которые всегда возвращают значение (число или строку символов), являющееся результатом расчетов или выполнения других операций. К функции обращаются по имени, за которым в круглых скобках указывают ее аргументы. В Access определены различные типы функций: статистические, математические, даты/время, текстовые и др. *Идентификаторы* – это ссылки на объекты БД, поля таблиц, элементы управления в формах и отчетах, их свойства. *Константы* – это фактические значения в виде числа, текстовой строки, даты (100, "Минск", #1-января-99#). *Оператор* – это операции над вышеперечисленными компонентами. В выражениях используются четыре типа операторов:

- арифметические: + (сложение), – (вычитание), * (умножение), / (деление), % (проценты), ^ (возведение в степень) и др.;
- операторы сравнения: = (равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), <> (не равно);
- логические операторы: AND (логическое умножение), OR (логическое сложение), NOT (логическое отрицание);
- оператор сцепления строк – &.

Пример выражения: =Sum([Закупочная цена])*0.08. В этом примере Sum() – это встроенная функция, [Закупочная цена] – идентификатор (ссылка на поле таблицы), * — арифметический оператор, а 0.08 – константа.

Пользователь имеет возможность создавать выражения с помощью Построителя выражений или путем набора их на клавиатуре.

Глоссарий

Выражение – комбинация некоторых или всех компонентов: функций, идентификаторов, констант, операторов.

Идентификатор – ссылка на объект БД, поле таблицы, элемент управления в форме, отчете, их свойства.

Константа – не изменяющееся значение.

Лента – набор вкладок, на которых расположены группы команд, упорядоченных по выполняемым функциям.

Макрос – набор из одной или нескольких макрокоманд, каждая из которых выполняет определенное действие.

Мастер – инструментальное средство, дающее возможность пошагового создания объектов БД или выполнения специальных операций.

Конструктор – инструментальное средство для самостоятельного создания объектов БД.

Модуль – программа для работы с БД, написанная на языке VBA.

Объекты БД – запросы, формы, отчеты, макросы, модули.

Оператор – операция над компонентами выражениями.

Построитель выражений – инструментальное средство для создания выражений.

Файл БД – файл, в котором хранятся все объекты БД.

Функция – небольшая программа, которая всегда возвращает значение (число или строку символов), являющееся результатом расчетов или выполнения других операций.

Элемент управления – графический объект, используемый при конструировании формы, отчета, страницы доступа к данным для отображения данного из базы, выполнения действия или в качестве украшения.

16. ВВЕДЕНИЕ В ЯЗЫК SQL

16.1. Язык SQL в СУБД

Язык **SQL** – структурированный язык запросов. Разработан IBM в 1970. Реализован в СУБД Oracle в 1976. С течением времени превратился в мощное средство для работы с *реляционными* БД. Существуют стандарта языка SQL: SQL-86 (SQL-1), SQL-92 (SQL-2), SQL:1999 (SQL-3 и др.

Язык SQL играет в СУБД важную роль – это язык: запросов, программирования, администрирования, распределения, шлюзов БД; доступа к данным в среде Internet; для связи клиентского приложения с сервером БД в архитектуре клиент/сервер;

Достоинства SQL: независимость от конкретных СУБД, переносимость с одной аппаратной среды в другую, наличие стандартов, реляционная основа.

16.2. Структура команды, типы данных и выражения в SQL

Каждая команда начинается с *глагола*, задающего действие, например, INSERT(добавить). После действия может следовать одно или несколько *предложений*, описывающих данные или уточняющих действие. Предложения начинаются с ключевых слов, например, WHERE (где), FROM (откуда) и др.

Пример SQL-команды:

DELETE FROM СВЕДЕНИЯ WHERE НОМ=980101

Типы данных, имеющиеся, например в SQL-2, приведены табл. 16.1.

Таблица 16.1.

Типы данных в SQL-2

Тип данных	Описание
CHAR(длина)	<i>Строки символов постоянной длины</i>
VARCHAR(длина)	Строки символов переменной длины
SMALLINT	Малые целые числа
INT	Целые числа
DEC(точн., степень)	Числа с фиксированной запятой
FLOAT(точность)	Числа с плавающей запятой

REAL	- - низкой точности
DOUBLE PRECISION	- - высокой точности
DATE	Дата

Выражения – это определенная последовательность полей БД, констант, функций, соединенных операторами. Существуют константы: с фиксированной запятой (62.3), с плавающей запятой (1.5E7), строковые ('Минск'), отсутствующее значение (Null). Операторы есть:

1) арифметические: +, -, *, /;

2) реляционные: =, >, <, >=, <=, <>;

3) логические: AND, OR, NOT;

4) булевы: IN – определяет множество, которому может принадлежать значение; BETWEEN – задает границы, в которые должно попадать значение; LIKE – для поиска по шаблону; IS NULL – для поиска NULL-значений.

Пример выражения: Цена * (Остаток_на_начало + Приход - Расход)

16.3. Функциональные возможности языка SQL

Определение данных

Язык определения данных (**DDL**) служит для создания и изменения структуры БД. Его ядро составляют команды: **CREATE TABLE**, **DROP TABLE**, **ALTER TABLE**.

Команда **CREATE TABLE** создает структуру таблицы. В ней указывается имя и тип данного в столбце таблицы. Могут задаваться свойства данного, например: **DEFAULT** – значение по умолчанию; **NOT NULL** – обязательность заполнения; **CHECK** – ограничение на значения; **UNIQUE** – условие уникальности; **PRIMARY KEY**, **FOREIGN KEY** – соответственно первичный ключ, внешний ключ.

Пример. Описать структуру таблицы:

СТУДЕНТ

НОМ	ФИО	ПОЛ
-----	-----	-----

Эта задача реализуется с помощью следующей SQL-команды:

```
CREATE TABLE СТУДЕНТ (НОМ INT PRIMARY KEY, ФИО CHAR(20)  
UNIQUE, ПОЛ CHAR(3))
```

Команда **DROP TABLE** удаляет таблицу.

Пример. **DROP TABLE** СТУДЕНТ

Команда **ALTER TABLE** изменяет структуру таблицы. В ней могут задаваться действия: **ADD** – добавить столбец; **DROP** – удалить столбец; **ALTER** – изменить значение по умолчанию и др.

Пример. В таблицу СТУДЕНТ добавить новое поле СЕМ_ПОЛ. Реализация на языке SQL:

```
ALTER TABLE СТУДЕНТ ADD СЕМ_ПОЛ CHAR(9)
```

Внесение изменений в БД

Язык манипулирования данными (**DML**) включает команды: **INSERT**, **UPDATE**, **DELETE**.

Команда **INSERT** добавляет новую строку в таблицу.

Пример. В таблицу СТУДЕНТ добавить строку со значениями:

НОМ – 555555, ФИО – СТРИЖ О. О., ПОЛ – ЖЕН

Реализация на языке SQL:

```
INSERT СТУДЕНТ (НОМ, ФИО, ПОЛ) VALUES (555555, ‘СТРИЖ О. О.’,  
‘ЖЕН’)
```

Команда **UPDATE** обновляет данные таблицы.

Пример. Изменить значение ФИО на БРЕЛЬ О. О. у студента с номером зачетки 555555. Реализация на языке SQL:

```
UPDATE СТУДЕНТ SET ФИО = ‘БРЕЛЬ О. О.’ WHERE НОМ =555555
```

Команда **DELETE** удаляет из таблицы строки.

Пример. Удалить из таблицы СТУДЕНТ сведения о студенте с номером зачетки 555555. Реализация на языке SQL:

```
DELETE FROM СТУДЕНТ WHERE НОМ =555555
```

Извлечение данных из БД

Оно возможно по команде **SELECT**. Это мощная команда, содержащая предложения, начинающиеся с ключевых слов: **FROM, WHERE, GROUP BY, HAVING, ORDER BY**. Она позволяет: извлекать данные из многих таблиц; производить вычисления над данными, в том числе с использованием статистических функций; создавать вложенные запросы и др.

Пример. Выбрать сведения о студентах мужского пола в алфавитном порядке. Реализация на языке SQL:

```
SELECT * FROM СТУДЕНТ WHERE ПОЛ= 'МУЖ' ORDER BY ФИО ASC
```

Управление транзакциями

Транзакция – несколько последовательных команд SQL, которые рассматриваются как единое целое. Для управления транзакциями существуют команды: **COMMIT** – успешное завершение транзакции, **ROLLBACK** – неуспешное (отменить транзакцию).

Пример. Студентка с номером зачетки 777777 вышла замуж – требуется изменить ее Ф.И.О. на *ГИЛЬ В. В.* и семейное положение на *замужем*. Реализация на языке SQL:

```
UPDATE СТУДЕНТ SET СЕМ_ПОЛ='ЗАМУЖЕМ' WHERE НОМ =777777  
UPDATE СТУДЕНТ SET ФИО = 'ГИЛЬ В. В.' WHERE НОМ =777777
```

Если данные две команды выполнены, как одна транзакция, успешно, то необходимо ввести команду **COMMIT WORK**, в противном случае – **ROLLBACK WORK**.

Управление доступом к данным

В языке SQL существуют *привилегии* – права пользователя на проведение тех или иных действий над определенным объектом БД: **SELECT** – право читать, **INSERT** – право добавлять, **UPDATE** – право обновлять, **DELETE** – право удалять и др. Назначение прав осуществляется по команде **GRANT**, а их отмена – по команде **REVOKE**.

Пример. Пользователю с именем ЗАМ_ДЕКАНА предоставить право на чтение и обновление таблицы СТУДЕНТ. Реализация на языке SQL:

```
GRANT SELECT, UPDATE ON СТУДЕНТ TO ЗАМ_ДЕКАНА
```

Пример. Пользователя с именем ЗАМ_ДЕКАНА лишить права на обновление таблицы СТУДЕНТ. Реализация на языке SQL:

REVOKE UPDATE ON СТУДЕНТ TO ЗАМ_ДЕКАНА

Встраивание SQL в прикладные программы

Для пересылки данных из БД в прикладную программу (на языке COBOL, BASIC, Perl, Pascal, C++) существуют следующие SQL-команды:

- **DECLARE** – определяет курсор для запроса. *Курсор* – это переменная для хранения результата запроса;
- **OPEN** – открывает курсор для чтения результата запроса;
- **FETCH** – считывает данные запроса в переменную прикладной программы;
- **CLOSE** – прекращает доступ к таблице результатов запроса.

16.4. Диалекты SQL в СУБД

С развитием СУБД их разработчики вносят в язык SQL изменения, дополнения – в результате появляются диалекты языка SQL. Их примеры:

- **PL/SQL** – в СУБД Oracle;
- **Transact-SQL** – в СУБД Microsoft SQL Server;
- **Informix-SQL** – в СУБД Informix;
- **Jet-SQL** – в СУБД Microsoft Access.

Глоссарий

SQL-транзакция – несколько последовательных команд SQL, которые должны выполняться как единое целое.

Выражение – определенная последовательность полей, констант, функций, соединенных операторами.

Диалект SQL – реализация языка SQL в конкретной СУБД со свойственными ей особенностями по сравнению со стандартом языка SQL.

Права – действия с объектом БД которые может выполнять пользователь.

Язык SQL – структурированный язык запросов к реляционным БД.

17. СИСТЕМЫ ОБРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ БД

17.1. Эволюция концепций обработки данных

Обработка данных – это совокупность методов и средств, осуществляющих преобразование данных. В своем развитии обработка прошла следующие четыре этапа: обработка на мэйнфреймах в пакетном режиме, в многотерминальных системах, на автономных персональных компьютерах (ПК), с использованием компьютерных сетей (КС).

Обработка на мэйнфреймах в пакетном режиме

Пользователь составлял *пакет заданий на языке описания заданий* на выполнение определенных операций над исходной программой и/или счет по программе, подготовленный на перфокартах. Затем передавал его на вычислительный центр (ВЦ) на обработку на мэйнфрейме, где оператор *без участия пользователя* последовательно вводил пакеты. Мэйнфрейм обрабатывал их и распечатанные результаты пользователь получал обычно на следующий день.

Обработка в многотерминальных системах

К мэйнфрейму, расположенному на ВЦ, подключались терминалы, расфронтенные по всему предприятию. **Терминал** – это устройство или набор устройств, предназначенных для взаимодействия с вычислительной системой или сетью ЭВМ. На мэйнфрейме устанавливались: БД, СУБД, прикладные программы (ПП), ОС_{ус} и ОС_{уд} – часть ОС, отвечающая соответственно за управление связью и за управление данными. Пользователи с помощью терминалов передавали запросы к БД. ОС_{ус} принимала их и передавала соответствующим ПП. Те обращались к СУБД, и она выполняла операции с БД, используя ОС_{уд}. Результаты запросов возвращались пользователям у терминалов.

Обработка данных на автономных ПК

На ПК пользователя устанавливалась СУБД, с помощью которой пользователь создавал на этом же ПК свою, локальную БД и работал с ней монопольно

(в однопользовательском режиме). Такая СУБД называется *настольной*. Она отвечает за выполнение запросов и за поддержание целостности БД.

Обработка данных с использованием КС

Обработка данных с использованием КС заключается в обмене данными между компьютерами посредством линий связи и специального коммуникационного оборудования. Обработка данных с использованием КС в настоящее время – самая реализуемая концепция обработки данных. В КС возможна *централизованная* и *децентрализованная* обработка данных.

17.2. Системы совместного использования файлов

Архитектура с совместным использованием файлов получила название «архитектура файл/сервер». В ней компьютеры объединены в сеть. На *файловом сервере* сети устанавливается ОС файлового сервера, включающая подсистемы ОС_{ус} и ОС_{уд}, и размещается БД. На *рабочих станциях* находятся: ПП, настольная СУБД, ОС_{ус}.

СУБД на рабочей станции запрос пользователя или ПП посылает к БД на сервере. ОС_{уд} выбирает из БД необходимые для его выполнения *таблицы целиком*. Затем ОС_{ус} пересылает их на рабочую станцию, где СУБД выполняет запрос.

Недостатки архитектуры следующие.

1. По сети передается гораздо больший объем данных, чем реально нужно для выполнения запроса. Вследствие этого сеть сильно перегружается.

2. Пользователи могут формировать запросы и на внесение изменений в БД. При этом блокируются записи, которые изменяются одним из пользователей, чтобы в это время другой пользователь не внес изменений в те же данные. Из-за этого системы с совместным использованием файлов редко используются для обработки БД больших объемов.

3. В архитектуре файл/сервер вся тяжесть выполнения запросов и управления целостностью БД ложится на СУБД пользователей, что является причиной не только «заторов» в сети, но и невысокой безопасности работы. Секретность

и конфиденциальность информации обеспечить также трудно.

Популярные настольные СУБД:

- **dBase и Visual dBase** (компания Ashton-Tate, разработана в начале 80-х годов прошлого столетия);
- **Paradox** (компания Ansa Software, в 1985г.). Последние версии являются составной частью Corel Office Professional;
- **Microsoft FoxPro и Visual FoxPro**. Происходят от настольной СУБД FoxBase фирмы Fox Software, которая впоследствии была приобретена Microsoft. версии FoxPro, начиная с 3.0, получили название Visual FoxPro.
- **Microsoft Access**. Появилась в начале 90-х годов. Это была первая настольная реляционная СУБД для 16-разрядной версии Windows. Популярность Access значительно возросла после включения в состав MS Office.

Достоинства настольных СУБД:

- просты для освоения и использования;
- обладают дружелюбным GUI;
- ориентированы на класс самых распространенных компьютеров – ПК и на самую широкую категорию пользователей – непрофессионалов;
- обеспечивают хорошее быстродействие при работе с небольшими БД.

Недостатки настольных СУБД стали проявляться с увеличением объемов БД и числа их пользователей и выразились в снижении производительности и возникновении сбоев при обработке данных.

17.3. Клиент/серверные системы

Базовый принцип архитектуры клиент/серверной системы – *централизация хранения и обработки данных*.

На *сервере сети* устанавливаются: операционная система; база данных; мощная серверная СУБД или **сервер БД** – программный компонент, обеспечивающий хранение больших объемов информации, ее обработку и представление пользователям в сетевом режиме; на клиентских компьютерах – клиентские приложения и операционная система для рабочих станций. В роли клиентских

компьютеров и сервера сети могут выступать мэйнфреймы, мини-ЭВМ, ПК.

Клиентское приложение формирует SQL-запрос к БД на сервере. *Сервер БД* интерпретирует запрос, выполняет его, формирует результат запроса и пересылает его по сети на клиентский компьютер. *Клиентское приложение* интерпретирует результат запроса необходимым образом и представляет его пользователю. Оно может также посылать запрос на обновление БД, и сервер БД внесет в нее необходимые изменения.

Преимущества архитектуры клиент/сервер следующие.

1. Клиенту по сети передается только результат запроса, поэтому в сети практически не возникает «заторов».

2. Обработка данных ложится в основном на сервер, который мощнее клиентов и поэтому способен быстрее обслуживать запросы. Как следствие этого, уменьшается потребность клиентских приложений в оперативной памяти.

3. Поскольку серверы способны хранить большие объемы данных, то на компьютерах-клиентах освобождается значительный объем дискового пространства для других целей.

4. При обращении одновременно нескольких клиентов к одним и тем же данным сервер БД разделяет выполнение их запросов во времени. Существенно повышается степень безопасности БД, так как правила целостности данных определяются сервером БД и являются едиными для всех приложений, использующих эту БД.

В архитектуре клиент/сервер существует *разделение функций между клиентскими приложениями и сервером БД*. *Сервер БД* помимо вышеописанных функций также:

- управляет целостностью БД;
- обеспечивает одновременную безопасную, отказоустойчивую многопользовательскую работу с одними и теми же данными;
- выполняет хранение и резервное копирование данных;
- выполняет бизнес-правила. Бизнес-правила реализуются с помощью средств СУБД: *ограничений, утверждений, хранимых процедур и триггеров*.

В реляционной СУБД примером *ограничений* могут быть ограничения на значения поля таблицы. Более общим типом ограничений являются *утверждения*, которые могут затрагивать более чем одну таблицу. Для реализации сложных бизнес-правил в СУБД используются хранимые процедуры и триггеры. **Хранимая процедура** – это именованный набор команд SQL, хранимый в БД на сервере. Хранимые процедуры выполняются сервером БД и могут: определяться относительно одной или более таблиц БД, читать и изменять данные таблиц БД, вызывать друг друга, вызываться из клиентского приложения, работающего с БД. **Триггер** – это хранящийся в БД на сервере именованный набор команд SQL, который автоматически выполняется, когда происходит некоторое заданное событие. В отличие от хранимой процедуры триггер выполняется автоматически, как реакция на заданное событие, и не может быть вызван из клиентского приложения.

Современные серверы БД должны удовлетворять следующим требованиям: масштабируемость, доступность, надежность, управляемость, наличие средств защиты данных от потери и несанкционированного доступа, поддержка доступа к данным с помощью Web-служб, поддержка стандартных механизмов доступа к данным. Как правило, серверы БД существуют для нескольких платформ, а часто и в разных редакциях. Наиболее популярные из них указаны в нижеприведенной таблице.

Сервер БД	Производитель
Oracle	Oracle Corp.
Microsoft SQL Server	Microsoft
Informix	Informix
Adaptive Server Enterprise	Sybase
DB2 Universal Database	IBM

17.4. Системы обработки распределенных БД

В распределенной БД (РабД) данные распределены по узлам КС. Каждый узел имеет собственную БД и может обращаться к данным, хранящимся на других узлах. Пользователь РабД не обязан знать, как ее компоненты размещены в узлах сети и представляет себе эту БД как единое целое. **РабД** – это совокуп-

ность логически взаимосвязанных БД, распределенных в КС. Распределенная СУБД (РаСУБД) – это программная система, обеспечивающая управление РаБД и прозрачность ее распределенности для пользователей. Информация о местоположении каждой из частей РаБД находится в глобальном словаре данных, который хранится на одном из компьютеров сети или может быть распределенным.

В общем случае все РаБД можно разделить на гомогенные и гетерогенные. В *гомогенной* (однородной) все локальные БД имеют одинаковые модели данных и управляются СУБД одного типа. В *гетерогенной* (неоднородной) локальные БД могут базироваться на разных моделях данных и управляться СУБД разных типов.

В системах обработки РаБД с целью приближения данных к месту их использования используется специальная операция – *репликация БД*, то есть, создаются копии БД в нескольких узлах сети, называемые *репликами*. После этого и в оригинал БД, и в реплики могут вноситься изменения. Затем выполняется операция синхронизации реплики с оригинальной БД – перенос в нее всех изменений из реплик.

Стратегии распределения данных по узлам сети:

- *разделения – нереплицирования.* БД распределена по компьютерам, реплики не создаются;
- *неразделения – реплицирования.* БД не разделена и создается реплика на каждом компьютере;
- *разделения – реплицирования.* БД распределена по компьютерам и на них создаются копии отдельных частей БД.

В РаБД на основе реляционной модели данных *фрагментация* (разделение) может быть вертикальной, горизонтальной и смешанной. При *горизонтальной* таблица разбивается на совокупности строк (записей), которые располагаются на различных компьютерах. При *вертикальной* таблица разделяется на совокупности столбцов (полей), которые хранятся на различных компьютерах. При *смешанной* происходит разделение реляционной таблицы как по

столбцам, так и по строкам.

К. Дейт сформулировал требования к РаБД и РаСУБД (12 правил):

- локальная автономность;
- никакой конкретный сервис не должен возлагаться на какой-либо специально выделенный центральный узел;
- непрерывность функционирования;
- независимость от местоположения, от фрагментации, от тиражирования (репликации);
- распределенная обработка запросов;
- управление распределенными транзакциями;
- независимость от оборудования, от операционных систем, от сети, от СУБД.

По К. Дейту *система обработки РаБД* – это слабосвязанная сетевая структура, на узлах которой располагаются локальные БД, которые автономны, независимы и доступ к ним обеспечивается, в общем случае, различными СУБД. Узлы обмениваются между собой потоками данных.

Некоторые прототипы РаСУБД:

- система **SDD-1** (разработана в конце 70-х – начале 80-х г. прошлого столетия, фирма Computer Corporation of America);
- система **System R*** (в начале 80-х г., фирма IBM);
- система **Distributed INGRES** (в начале 80-х г., в Калифорнийском университете в Беркли).

В настоящее время в большинстве сетевых СУБД предусмотрены отдельные виды поддержки РаБД с различным набором функций. Например, они есть в **ORACLE, DB2**.

Достоинства РаБД: более полно отражают территориально распределенную структуру предприятия и обеспечивают большую живучесть ИС, так как в случае разрушения одной из локальных БД другие БД остаются работоспособными. *Недостаток* РаБД – повышенная сложность их практической реализации.

17.5. Типы интерфейсов доступа к данным базы

Доступ к данным базы возможен: посредством форм, с помощью языка запросов (важнейший из всех языков запросов – язык SQL), посредством хранимых процедур и триггеров, через прикладные программы, написанные на языках программирования COBOL, BASIC, Perl, Pascal, C++. Некоторые из них пишутся на встроенных в СУБД языках, как например, xBase.

17.6. Хранилище данных

В последнее время в системах поддержки принятия решений широко используются *OLAP-технология и хранилища данных*.

OLAP – *оперативная аналитическая обработка*. 12 определяющих принципов OLAP сформулировал в 1993 г. Е. Ф. Кодд. Позже они были переработаны в тест FASMI, требующий, чтобы OLAP-приложение предоставляло возможности *быстрого анализа разделяемой многомерной информации*, накопленной в хранилище.

Хранилище данных (ХД) – это ориентированная на предметную область, интегрированная, статичная, не разрушаемая совокупность данных, предназначенная для поддержки принятия решений. *Предметная ориентированность* означает, что ХД предназначено для предоставления данных, связанных с *одним* организационным процессом; *интегрированность* – применение единых законов именования, способов вычисления значений данных, полученных из различных источников; *статичность* – ХД содержит как исторические данные (до 5 лет), так и данные, которые имели статус текущих при последнем погружении в ХД; *неразрушаемая совокупность данных* – в ХД помещается большой объем данных, которые будучи раз погруженными, уже никогда больше не подвергаются каким-либо изменениям.

Сравнительная характеристика ХД и БД приведена в табл. 17.1.

Таблица 17.1.

Сравнение хранилищ данных и баз данных

Базы данных	Хранилища данных
Используются	
<i>сотрудниками</i>	<i>высшим руководством</i>
Поддерживают	
<i>тактические</i> направления развития бизнеса	<i>стратегическое</i> направление развития бизнеса
Используются	
для <i>обработки</i> данных	для <i>анализа</i> данных
Ориентированы	
на <i>приложения</i>	на <i>предметную область</i>
Хранят	
только <i>текущие</i> данные	<i>исторические</i> данные
Допускают	
только <i>предсказуемые</i> запросы пользователей	<i>непредсказуемые</i> запросы пользователей

Данные в ХД попадают прямо или косвенно *из оперативных систем* (OLTP-систем), которые предназначены для автоматизации бизнес-процессов. Кроме того, ХД могут пополняться *за счет внешних источников*, например, статистических отчетов.

Классификация ХД по Б. Инмону (по отраслевому принципу применения ХД): финансовые ХД; ХД в области страхования; ХД для управления персоналом; ХД, поддерживающие технологию обнаружения новых данных; ХД в области телекоммуникаций; глобальные ХД (для глобального представления деятельности организации).

Существует несколько вариантов реализации ХД:

- *виртуальное ХД* – архитектура обеспечивает доступ к данным в режиме реального времени через специальное программное обеспечение.
- *киоски данных (витрины данных)* – небольшое ХД, обслуживающее одно из *направлений* бизнеса компании или одно ее структурное подразделение

- *глобальное ХД* – архитектура представляет собой единый источник интегрированных данных организации;
- *ХД с многоуровневой архитектурой (корпоративное ХД)* – архитектура является разновидностью глобального ХД, в котором технологически реализуется три уровня: на 1-м – корпоративное ХД; на 2-м – связанные киоски данных на основе многомерной СУБД; на 3-м – клиентские приложения пользователей со средствами анализа данных;
- *встроенное ХД* – это ХД, которое встраивается в виртуальное предприятие или используется как компонент аналитической поддержки в информационной реализации бизнес-функций.

Характеристика программного обеспечения для разработки ХД ведущих производителей приведена в табл. 17.2.

Таблица 17.2

Программное обеспечение для разработки ХД

Производитель	Название программного продукта	СУБД для ХД	Язык обработки данных
IBM	Data Warehouse Plus	Семейство объектно-реляционных СУБД DB2	SQL
Oracle	Warehouse Technology Initiative	Семейство объектно-реляционных СУБД Oracle 7/8i/9i/10g	SQL, дополненный набором функций для аналитической обработки данных
Sybase	Warehouse Works	реляционная СУБД Sybase System II	SQL
Microsoft	Data Warehousing Framework	реляционная СУБД MS SQL	SQL со встроенными средствами

		Server 2003	обработки мно- гомерных кубов
--	--	--------------------	----------------------------------

Глоссарий

Гетерогенная распределенная БД – распределенная БД, фрагменты которой – локальные БД – основаны на разных моделях данных и управляются различными типами СУБД .

Гомогенная распределенная БД – распределенная БД, фрагменты которой – локальные БД – имеют одинаковые модели данных и управляются однотипной СУБД.

Децентрализованная обработка – обработка информации, базирующаяся на решении локальных задач и работе с локальными БД на рабочих местах пользователей.

Исходная программа – программа, написанная на алгоритмическом языке.

Киоск данных – небольшое хранилище данных, обслуживающее одно из направлений бизнеса компании или одно ее структурное подразделение.

Клиентский компьютер – компьютер, генерирующий запросы к другим компьютерам или пользующийся ресурсами, которые доступны на различных серверах.

Клиентское приложение – приложение, установленное на компьютере-клиенте.

Настольная СУБД – СУБД, предназначенная для создания локальной БД и работы с ней на персональном компьютере.

Обработка данных – совокупность методов и средств, осуществляющих преобразование данных.

Распределенная БД – совокупность логически взаимосвязанных БД, распределенных в компьютерной сети.

Распределенная СУБД – программная система, которая обеспечивает управление распределенной БД и прозрачность ее распределенности для пользователей.

Сервер – специальный компьютер сети, настроенный на выполнение определенного набора функций.

Сервер БД – программный компонент, обеспечивающий хранение больших объемов информации, ее обработку и представление пользователям в сетевом режиме.

Терминал – устройство или набор устройств, предназначенных для взаимодействия с вычислительной системой или сетью ЭВМ.

Триггер – хранящийся в БД на сервере именованный набор команд SQL, который автоматически выполняется, когда происходит некоторое заданное событие.

Файловый сервер – компьютер в сети, обеспечивающий хранение файлов и предоставляющий санкционированный доступ пользователям и приложениям, исполняемым на других компьютерах сети.

Фрагментация данных – технология разбиения БД на составные части – фрагменты, хранимые в различных узлах сети.

Хранилище данных – ориентированная на предметную область, интегрированная, статичная, не разрушаемая совокупность данных, предназначенная для поддержки принятия решений.

Хранимая процедура – именованный набор команд SQL, хранимый в БД на сервере.

18. АДМИНИСТРИРОВАНИЕ БАЗЫ ДАННЫХ

18.1. Функции администратора базы данных

С многопользовательскими БД связан ряд следующих проблем.

1. Многопользовательские БД сложны в проектировании и разработке, так как при этом существует множество перекрывающихся пользовательских представлений.

2. Изменения, вносимые со временем в структуру БД, должны тщательно контролироваться, чтобы изменения, сделанные для одной группы пользователей, не вызвали проблем для другой.

3. При параллельной обработке запросов нескольких пользователей надо принимать специальные меры, чтобы действия одного пользователя не оказали непредусмотренного влияния на действия другого пользователя.

4. Необходима разработка системы безопасности, которая позволит выполнять только строго определенные действия и только пользователям, имеющим для этого достаточные полномочия.

5. Со временем требуется модифицировать СУБД или внедрять новую.

Для решения этих задач выполняется *администрирование БД* – это обеспечение надежного и эффективного функционирования БД, а также адекватности ее содержания информационным потребностям пользователей, отображения в БД актуального состояния предметной области. Оно включает администрирование: предметной области, БД, приложений, безопасности данных. Им занимается администратор БД, функции которого в общем случае следующие:

- контроль целостности и восстановление БД;
- реорганизация БД в соответствии с изменениями предметной области;
- подключение новых разработчиков и пользователей, назначение паролей и прав доступа к конкретным данным;
- настройка СУБД для эффективной обработки данных и обслуживания пользователей;

- консультирование пользователей по особенностям используемой версии СУБД;
- разработка документации, регламентирующей действия пользователей по отношению к БД;
- контроль изменения объема БД и определение целесообразности модернизации оборудования;
- мониторинг и оптимизация производительности БД.

Роль администратора БД очень важна для субъектов хозяйствования.

18.2. Защита баз данных

Причины разрушения и потери данных базы различны. Это: сбои оборудования; физические воздействия на аппаратные средства БД; стихийные бедствия; ошибки санкционированных пользователей; умышленные вредоносные действия несанкционированных пользователей или программ; программные ошибки СУБД или операционных систем; ошибки в прикладных программах; совместное выполнение конфликтных запросов пользователей и др.

Защита БД должна решать две основные задачи: обеспечение *безопасности* данных и обеспечение *секретности* данных.

Средства для защиты БД в СУБД: защита паролем, шифрование данных, разграничение прав доступа. Если этих средств недостаточно, то разрабатываются специальные программные продукты по защите информации, и они интегрируются с СУБД.

Главное достоинство защиты с помощью *пароля* – простота и привычность. Требования к паролю: он должен состоять из комбинации букв, цифр, специальных знаков, быть не менее чем из шести символов, не должен содержать пробелы, часто изменяться.

Шифрование – это преобразование исходных данных по специальным алгоритмам в новое представление, скрывающее содержание исходной информации. **Дешифрование** – это обратный шифрованию процесс. При шифровании БД ее файл кодируется и становится недоступным для просмотра инфор-

мации с помощью служебных программ.

Возможны два *режима работы* с зашифрованными БД:

- 1) информация дешифруется на внешнем носителе. Затем с ней производятся необходимые действия, после чего для записи на внешней носитель она снова зашифровывается;
- 2) СУБД выполняет дешифрование в оперативной памяти непосредственно перед выполнением конкретных действий с данными. При этом процедуры шифрования встраиваются в СУБД.

Каждый из режимов имеет свои достоинства и недостатки. Любое шифрование снижает уровень производительности СУБД.

Разграничение прав доступа – необходимая функция любой многопользовательской СУБД. Администратор БД настраивает права доступа пользователей в соответствии с их служебными обязанностями с помощью средств СУБД. В современных СУБД поддерживаются концепции – *пользователи* и *группы* и можно предоставить им права доступа к определенным объектам БД – от *только чтения* до *реорганизации всей БД*.

18.3. Правовая охрана баз данных

Безопасность БД охраняется в нашей стране в законодательном порядке. БД, также как и компьютерная программа, приравнивается к литературному произведению и может быть объектом авторского права, что предусмотрено статьей 7 «Закона об авторском праве Республики Беларусь». Тогда ей предоставляется охрана гражданским, административным и уголовным законодательством.

18.4. Восстановление базы данных

Восстановление БД осуществляется в случае ее физического повреждения или нарушения целостности. Средства восстановления БД – резервное копирование БД и журнал изменений БД.

Резервная копия БД создается в момент, когда состояние БД является це-

лостным и на иных внешних устройствах, чем то, на котором располагается сама БД. Восстановление БД заключается в переносе на рабочее устройство резервной копии БД или поврежденной ее части и повторном проведении всех изменений, сделанных после создания копии и до момента возникновения сбоя.

Журнал изменений БД – это особая часть БД, недоступная пользователям СУБД, в которую поступают записи обо всех изменениях основной части БД. В большинстве современных реляционных СУБД журнал изменений называется *журналом транзакций*, в котором регистрируются в хронологическом порядке все изменения, вносимые в БД каждой транзакцией. При ведении журнала транзакций восстановить БД можно методом:

- 1) *накат* (раскрутка) – в сохраненную копию БД вносятся результаты всех завершенных транзакций. При этом транзакции не обрабатываются повторно, а производятся изменения в БД согласно записям в журнале транзакций;
- 2) *откат* – отменяются изменения, произведенные в БД ошибочными или незавершенными транзакциями. Затем повторно запускаются транзакции, которые выполнялись в момент возникновения сбоя.

18.5. Оптимизация работы базы данных

Оптимизировать работу БД позволяет:

- технология ускоренного доступа к данным БД с помощью индексов – *индексирование*. **Индекс** – это средство ускорения операции поиска записей в таблице, а также других операций, требующих поиска, извлечения, модификации, сортировки и др.;
- технология быстрого прямого доступа к записи БД на основе заданного значения ключевого поля записи – *хеширование*. Каждая запись БД размещается по *хеш-адресу*, который вычисляет СУБД с помощью *хеш-функции* на основе значения ключевого поля этой записи (*хеш-поля*);
- *технология сжатия на основе различий* – когда некоторое значение заменяется сведениями об его отличиях от предыдущего значения. Один из способов

применения такой технологии сжатия – это удаление повторяющихся символов в начале каждой записи с указанием их количества.

Пример:

До сжатия	После сжатия
Эколог.....	0 – Эколог
Экология.....	6 – ия
Экологический.....	7 – ческий

- *внутри файловое иерархическое сжатие.*

Пример:

Файл УСПЕВАЕМОСТЬ

Предмет	Ф.И.О	Оценка
Информатика	Брель Б.В.	5
Маркетинг	Гиль А.Б.	4
Маркетинг	Друздь Г.Д.	3
Политология	Смаль В.Г.	4

Результат внутри файлового *иерархического* сжатия:

Предмет	Ф.И.О	Оценка
Информатика	Брель Б.В.	5
Маркетинг	Гиль А.Б.	4
	Друздь Г.Д.	3
Политология	Смаль В.Г.	4

- *кодирование Хаффмана* – когда наиболее часто повторяющиеся символы кодируются битовыми строками меньшей длины.

Пример:

Символ	Частота появления, %	Код
А	45	1

В	40	01
Г	35	001
Б	30	0001

18.6. Возможности администрирования БД в СУБД Access

В СУБД Access имеются встроенные средства администрирования БД:

- просмотр сведений о БД;
- печать описания БД и ее объектов, произведенного системой Access в процессе их создания пользователем;
- экспорт/импорт объектов БД;
- преобразование БД в другие форматы;
- сжатие и восстановление БД;
- создание резервной копии БД;
- шифрование БД;
- сохранение БД в файле формата .accde с целью ее сжатия и защиты от изменения форм, отчетов и модулей;
- защита БД паролем;
- поддержка целостности данных.

Глоссарий

Администрирование БД – обеспечение надежного и эффективного функционирования БД, а также адекватности ее содержания информационным потребностям пользователей, отображения в БД актуального состояния предметной области.

Журнал изменений БД – особая часть БД, недоступная пользователям СУБД, в которую поступают записи обо всех изменениях основной части БД.

Журнал транзакций – особая часть реляционной БД, недоступная пользова-

телям СУБД, в которую поступают записи обо всех ее изменениях.

Индекс – средство ускорения операции поиска записей в таблице, а также других операций, требующих поиска, извлечения, модификации, сортировки и др.

Накат – внесение в сохраненную копию БД результатов всех завершенных транзакций.

Откат – отмена изменений, произведенных в БД ошибочными или незавершенными транзакциями.

Шифрование – преобразование исходных данных по специальным алгоритмам в новое представление, скрывающее содержание исходной информации.

19. БАЗЫ ЗНАНИЙ И МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

19.1. Базы знаний

Знания – это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области.

Знания *по степени научности* подразделяются на *житейские* (обыденные знания) и *научные*. Научные знания классифицируются на *поверхностные* (эмпирические) – знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области и *глубинные* (теоретические) – абстракции, аналогии, схемы, отображающие структуру и природу процессов, протекающих в предметной области.

Знания *по местонахождению* делят на *личностные (неявные, скрытые)* – знания людей и *формализованные (явные)* – например, знания в документах, на компакт-дисках, в Интернете.

Знания *по природе* делят на *процедурные* и *декларативные*. *Декларативные* знания содержат в себе лишь представление о неких понятиях. *Процедурные* знания имеют активную природу – определяют представления о средствах и путях получения новых знаний, проверки знаний. Это алгоритмы разного рода.

База знаний (БЗ) – это семантическая модель, описывающая предметную область и позволяющая отвечать на такие вопросы из этой предметной области, ответы на которые в явном виде не присутствуют в базе. БЗ включает один или несколько специальным образом организованных файлов, хранящих систематизированную совокупность понятий, правил и фактов, относящихся к некоторой предметной области. Их содержимое оформляется, связывается между собой и представляется так, что на его основе можно с помощью специ-

альных программ осуществлять рассуждения и делать выводы, получая сведения, которые в явном виде могут не присутствовать в базе знаний.

Базы знаний являются ядром интеллектуальных систем. Один из их видов – это экспертные системы. *Экспертная система* – это компьютерная программа, способная заменить специалиста-эксперта в решении проблемной ситуации. Рассмотрим основные модели представления знаний.

19.2. Продукционные модели

Продукционная модель – это модель, основанная на представлении знания в виде правил «Если (условие), то (действие)». «*Условие*» – некоторое предложение-образец, по которому осуществляется поиск в БЗ, а «*действие*» – действия, выполняемые при успешном исходе поиска.

В экспертной системе с БЗ, созданной по продукционной модели, исходные факты (данные) хранятся в *базе фактов (рабочей памяти)*, а набор правил – в БЗ. На основе базы фактов запускается *машина вывода* или *интерпретатор правил* – это программа, управляющая перебором правил из БЗ. Она состоит из 2-х компонентов – компонента вывода и управляющего компонента. Действие *компонента вывода* основано на применении правила *modus ponens*:

Если известно, что истинно утверждение А и существует правило вида «Если А, то В», тогда утверждение В также истинно.

Правила срабатывают, когда находятся факты, удовлетворяющие их левой части: если истинна посылка, то должно быть истинно и заключение.

***Управляющий компонент* определяет порядок применения правил и выполняет четыре функции:**

- 1) *сопоставление* – образец правила сопоставляется с имеющимися фактами;
- 2) *выбор* – если в конкретной ситуации может быть принято сразу несколько правил, то из них выбирается одно, наиболее подходящее по заданному критерию;
- 3) *срабатывание* – если образец правила при сопоставлении совпал с какими-либо фактами из рабочей памяти, то правило срабатывает;

4) *действие* – рабочая память подвергается изменению путем добавления в нее заключения сработавшего правила. Если в правой части правила содержится указание на какое-либо действие, то оно выполняется.

Достоинства продукционной модели: наглядность, высокая модульность, легкость внесения дополнений и изменений, простота механизма логического вывода; *недостаток* – при накоплении достаточно большого числа (порядка нескольких сотен) продукций (правил) они начинают противоречить друг другу. Примеры экспертных систем на основе продукционной модели – ЭКСПЕРТ, ЭКО и др.

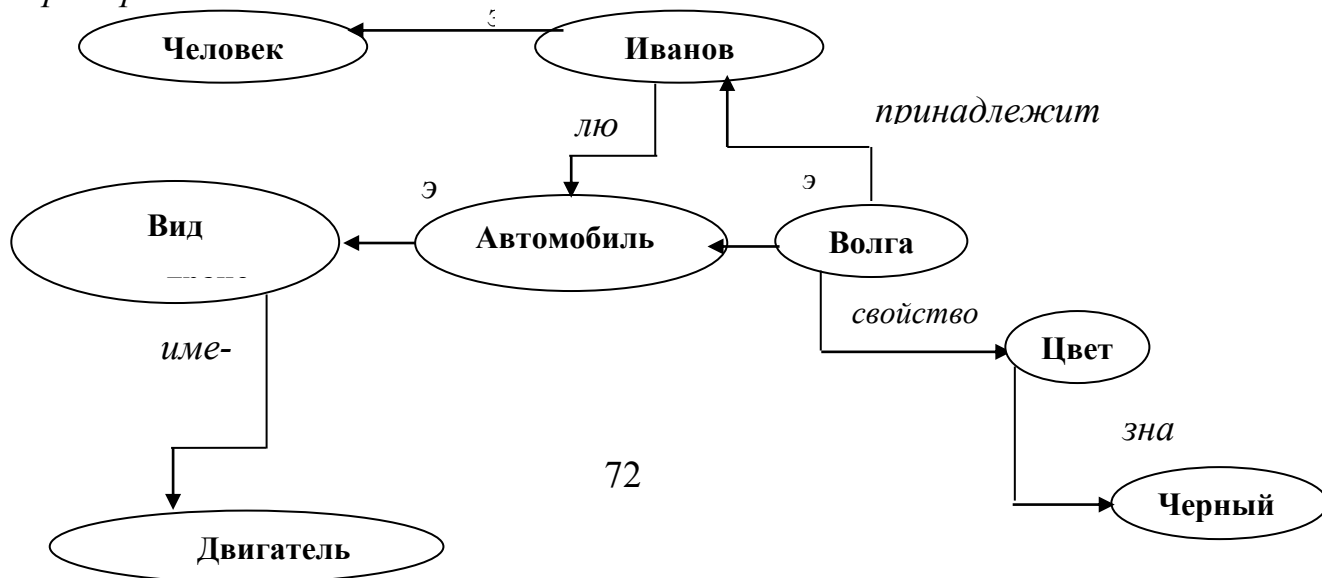
19.3. Семантические сети

Семантическая сеть – это ориентированный граф, вершины которого есть понятия, а дуги – отношения между ними. *Понятия* – абстрактные или конкретные объекты. Наиболее часто используемые *отношения*:

- 1) связи типа “часть-целое” (например, “класс-подкласс”);
- 2) функциональные связи (например, производит);
- 3) количественные (например, $>$, $<$, $=$);
- 4) пространственные (например, далеко от);
- 5) временные (например, раньше);
- 6) атрибутивные (например, иметь свойство);
- 7) логические (например, и, или, не) и др.

Поиск решения в БЗ типа семантической сети сводится к поиску фрагмента сети, отражающего поставленный запрос к базе.

Пример семантической



Преимущество – более других соответствует современным представлениям об организации долговременной памяти человека. *Недостаток* – сложность организации процедуры поиска вывода на семантической сети.

Экспертные системы, использующие семантические сети в качестве языка представления знаний, – **PROSPECTOR, CASNET, TORUS**.

Создавать БЗ по семантической сети позволяют программные продукты, называемые редакторами онтологий. *Онтология* – это формальное явное описание понятий в рассматриваемой предметной области и отношений между ними. Популярным редактором онтологий является **PROTÉGÉ**, разработанный в Стэнфордском университете США. Он позволяет:

- создавать иерархию классов;
- задавать свойства классов через определение слотов;
- создавать связи между классами;
- редактировать стандартные формы для заполнения слотов;
- создавать экземпляры классов и устанавливать отношения между ними;
- графически представлять разработанную онтологию;
- формировать, выполнять и сохранять запросы к БЗ, созданной на основе онтологии;

19.4. Фреймы

Фрейм – это абстрактный образ для представления некоего стереотипа восприятия. Все многообразие знаний о мире можно отобразить через:

- 1) *фреймы-структуры* – для обозначения объектов и понятий (заем, залог, вексель);
- 2) *фреймы-роли* (например, менеджер, кассир, клиент);
- 3) *фреймы-сценарии* (например, банкротство, собрание акционеров);

4) *фреймы-ситуации* (например, тревога, авария, рабочий режим устройства).

Различают *фреймы-образцы (фреймы-классы)*, хранящиеся в БЗ, и *фреймы-экземпляры*, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Любой фрейм-экземпляр содержит набор атрибутов – *слотов*, которым присваиваются значения. Каждый слот имеет уникальное имя.

Пример фрейма-экземпляра Лекция:

ПРЕДМЕТ	<i>Информатика</i>
ЛЕКТОР	Красилов И.И.
АУДИТОРИЯ	403
СЛУШАТЕЛИ	97

Существует несколько *способов получения слотом значений* во фрейм-экземпляре: по умолчанию от фрейма-образца; через наследование свойств от фрейма, указанного в слоте АКО (это); по формуле, указанной в слоте; через присоединенную процедуру, выполняемую при определенных условиях; из диалога с пользователем; из БД. В качестве значения слота может выступать имя другого фрейма. Тогда образуются *сети фреймов*.

Преимущества фреймовой модели: отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность; однородность представления знаний и возможность их типового текстового описания с помощью специальных языков представления знаний.

Примеры фрейм-ориентированных экспертных системы – **ANALYST**, **МОДИС**, **TRISTAN**, **ALTERID**.

19.5. Формальные логические модели

Формальная логика – это наука об общих структурах и законах правильного мышления, образования и сочетания понятий и высказываний, о правилах умозаключений независимо от их конкретного содержания.

Моделирование логики рассуждений происходит по схеме – из одной или нескольких посылок, которые считаются истинными, следует сделать логически верное заключение. И посылки, и заключение должны быть представлены на понятном языке, адекватно отображающем предметную область, в которой проводится вывод. В обычной жизни это наш естественный язык общения; в математике это язык определенных формул.

Логические выражения, построенные в данном языке, могут быть истинными или ложными. Некоторые из этих выражений, являющиеся всегда истинными, объявляются *аксиомами* или *постулатами*. Они составляют ту базовую систему посылок, исходя из которой и пользуясь определенными правилами вывода, можно получить заключения в виде новых выражений, также являющихся истинными.

Классическими примерами *аксиоматических (формальных)* систем являются исчисление высказываний и исчисление предикатов. **Исчисление высказываний** – это область математической логики, называемая булевой алгеброй. Она изучает высказывания и операции над ними. **Высказывание** – это предложение, которое может быть истинно или ложно. Наиболее известными операциями булевой алгебры являются конъюнкция и дизъюнкция. Так, *дизъюнкция высказываний* – новое высказывание, сконструированное из двух и более исходных высказываний, и истинное в тех случаях, когда истинно хотя бы одно из исходных высказываний. *Конъюнкция высказываний* – новое высказывание, сконструированное из двух и более исходных высказываний, и истинное в тех случаях, когда истинны все исходные высказывания.

В булевой алгебре ни структура высказываний, ни их содержание не затрагиваются. В то же время и в науке, и в практике используются заключения, существенно зависящие как от структуры, так и от содержания используемых в них высказываний. Возникает необходимость в расширении логики высказываний – в логике предикатов. *Логика предикатов* расчленяет элементарное высказывание на *субъект* (подлежащее) и *предикат* (сказуемое). **Субъект** – это то, о чем что-то утверждается в высказывании. **Предикат** – это то, что утвер-

ждается о субъекте. Например, в высказывании «7 – простое число», «7» – субъект, «простое число» – предикат. Если число 7 заменить переменной X из множества натуральных чисел N , то получим высказывание « X – простое число». При одних значениях X (например, $X=13$, $X=17$) результат этого высказывания истинный, а при других значениях X (например, $X=10$, $X=18$) – ложный (результат зависит от содержания). Здесь предикат становится функцией субъекта и выражает свойство субъекта.

Достоинство формальных систем – исчисления высказываний и исчисления предикатов – в том, что они хорошо исследованы и имеют прекрасно разработанные модели логического вывода. *Недостатки* – закрытость, негибкость. Поэтому формальные системы как модели представления знаний могут использоваться только в тех предметных областях, которые хорошо локализируются и мало зависят от внешних факторов. К тому же, очень высокие требования к предметной области – полнота и непротиворечивость «базового аксиоматического набора» – обусловили то, что в промышленных экспертных системах формальные логические модели практически не используются.

Глоссарий

Знания – закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области.

База знаний – семантическая модель, описывающая предметную область и позволяющая отвечать на такие вопросы из этой предметной области, ответы на которые в явном виде не присутствуют в базе.

Продукционная модель – модель, основанная на представлении знания в виде правил «Если (условие), то (действие)».

Семантическая сеть – ориентированный граф, вершины которого есть понятия, а дуги – отношения между ними.

Фрейм – формализованная модель для отображения образа.

Формальная логика – наука об общих структурах и законах правильного мыш-

ления, образования и сочетания понятий и высказываний, о правилах умозаключений независимо от их конкретного содержания.

Экспертная система – компьютерная программа, способная заменить специалиста-эксперта в решении проблемной ситуации.

20. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

20.1. Задания по теме «Моделирование базы данных в программе ERwin»

Задание 1. Ознакомиться с предметной областью согласно варианту, заданному преподавателем из числа ниже перечисленных.

Вариант 1.

Описание предметной области **ПОСТАВКА ИЗДЕЛИЙ**

Фабрика поставляет изделия заказчикам по договорам. Проектируется БД, которая будет использоваться для анализа отгрузки продукции. В БД планируется хранить информацию:

- об ИЗДЕЛИЯХ: код изделия, наименование изделия, цена изделия (тыс. руб.);
- ЗАКАЗАХ на поставку: код заказа, код изделия, плановая поставка (шт.);
- фактических ОТГРУЗКАХ: код отгрузки, код заказа, дата отгрузки, код изделия, отгружено изделия (шт.).

Между сущностями существуют следующие связи:

- ИЗДЕЛИЕ–имеет–ЗАКАЗ
- ИЗДЕЛИЕ–соответствует–ОТГРУЗКА

Вариант 2.

Описание предметной области **ТУРАГЕНСТВО**

Турагенство продает путевки путешествий по разным странам. В каждую страну организуется несколько маршрутов, организуемых с целью отдыха, экскурсии, лечения и др. Проектируется БД, которая позволит определять наиболее популярные маршруты, отслеживать обращения клиентов и др. В БД планируется хранить информацию:

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

- ПРОДАЖАХ: код маршрута, количество проданных путевок по маршруту, дата продажи.

Между сущностями существуют следующие связи:

- МАРШРУТ–организуется–СТРАНА
- МАРШРУТ–участвует–ПРОДАЖА

Вариант 3.

Описание предметной области **МАГАЗИН**

Магазин продает ноутбуки и средства связи от разных поставщиков. Проектируется БД, которая будет использоваться для анализа продаж в магазине. В БД планируется хранить информацию:

- о ТОВАРАХ: код товара, наименование товара, количество товара (шт.), цена товара (руб.);
- ПОСТАВЩИКАХ товаров: код поставщика, наименование поставщика, адрес, телефон;
- ПРОДАЖАХ товаров: код продажи, код товара, дата продажи, количество проданного товара (шт.).

Между сущностями существуют следующие связи:

- ТОВАР–поставляет–ПОСТАВЩИК
- ТОВАР–участвует–ПРОДАЖА

Вариант 4.

Описание предметной области **РЕАЛИЗАЦИЯ ТОВАРОВ**

Отделы торгового дома продают различные виды товаров и ведут учет сведений об их реализации. Проектируется БД, которая будет использоваться для анализа реализации товаров в отделах, определения наиболее востребованных товаров и др. В БД планируется хранить информацию:

- об ОТДЕЛАХ: код отдела, наименование отдела;
- ТОВАРАХ: код товара, наименование товара, единица измерения, цена товара (руб.);
- РЕАЛИЗАЦИИ: код товара, дата реализации, реализовано товара (руб.).

Между сущностями существуют следующие связи:

- ОТДЕЛ–продает–ТОВАР

- ТОВАР–имеет–РЕАЛИЗАЦИЯ

Вариант 5.

Описание предметной области **ПОЧТА**

Почта каждое полугодие осуществляет подписку на различные издания на один, три или шесть месяцев. Проектируется БД, которая будет использоваться для учета подписчиков и выписанных ими изданий. В БД планируется хранить информацию:

- об ИЗДАНИЯХ: индекс издания; вид издания (газета, журнал); название издания; стоимость подписки на издание на 1 месяц, руб.;
- ПОДПИСЧИКАХ: код подписчика, Ф.И.О. подписчика, адрес подписчика;
- ПОДПИСКАХ: код подписчика; индекс издания; срок подписки, в месяцах; месяц и год начала доставки издания.

Между сущностями существуют следующие связи:

- ПОДПИСЧИК–осуществляет–ПОДПИСКА
- ИЗДАНИЕ–получает–ПОДПИСКА

Задание 2. Определить тип и мощность связей между сущностями согласно их определению в программе **ERwin**.

Задание 3. Создать диаграмму *уровня сущности* (на вкладке с именем Сущности).

Задание 4. Создать *физическую модель* БД для СУБД Access (на вкладке с именем Физическая).

Задание 5. Создать диаграмму *уровня атрибутов* (на вкладке с именем Атрибуты).

Задание 6. Создать *логическую модель* БД (на вкладке с именем Логическая).

20.2. Задания по теме «Технологии работы с базой данных в СУБД Microsoft Access»

Задание 1. Создать БД согласно варианту, заданному преподавателем из числа ниже перечисленных. Для этого:

- определить тип связи между таблицами;
- создать структуру каждой таблицы;
- установить связь между таблицами;
- заполнить таблицы данными.

Вариант 1. Фирма шьет женские платья различных моделей и поставляет их торговым домам. С целью ведения учета отгрузки готовой продукции создать БД **МОДА**, включающую две связанные таблицы:

МОДЕЛЬ

Модель	Цена, тыс. р.
С/7	250
К-9	300

ОТГРУЗКА

Наименование торгового дома	Количество отгруженных изделий, шт.	Модель
Радуга	35	С/7
Радуга	20	К-9
Мечта	25	К-9

Вариант 2. Водители автокомбината совершают междугородние грузовые перевозки. С целью учета расхода бензина создать БД **АВТОКОМБИНАТ**, включающую две связанные таблицы:

ВОДИТЕЛЬ

Ф.И.О. водителя	Номер машины	Расход бензина по норме на 100 км, л
Смаль С.С.	52АМ-7	10
Гринь Г.Г.	11МР-6	12

ПЕРЕВОЗКА

Номер машины	Дата перевозки	Пробег машины, км
52AM-7	11.11.12	300
52AM-7	12.12.12	450
11MP-6	13.12.12	200

Вариант 3. В многоквартирных домах установлены общие газовые счетчики. С целью анализа оплаты за газ создать БД **ЖЭС**, включающую две связанные таблицы:

ОПЛАТА

Адрес (улица, дом)	Оплата за газ на 1 человека, р.
Брыля, 20	5000
Сапеги, 11	6000

КВАРТИРОСЪЕМЩИК

Ф.И.О.	Адрес (улица, дом)	Номер квартиры	Количество жильцов
Лис Л.Л.	Брыля, 20	5	2
Зыль В.В.	Брыля, 20	7	4
Кнель И.И.	Сапеги, 11	9	3

Вариант 4. Клубы города предоставляют посетителям различные услуги. С целью анализа финансового состояния клубов создать БД **ФИНАНСЫ**, включающую две связанные таблицы:

КЛУБ

Клуб	Адрес
Юность	ул. Садовая, 11
Досуг	ул. Минина, 11

СОСТОЯНИЕ

Клуб	Вид услуги	Доходы, млн. р.	Расходы, млн. р.
Юность	казино	100	20
Юность	видео показ	80	10
Досуг	игровые автоматы	90	30

Вариант 5. Банки совершают операции с валютой. С целью анализа деятельности банков создать БД **ВАЛЮТА**, включающую две связанные таблицы.

БАНК

Банк	Адрес
Олимп	ул. Победы, 15
Талер	ул. Лесная, 27

ПРОДАЖА

Банк	Месяц	Вид валюты	Продано на сумму, млн. р.
Олимп	май	евро	700
Олимп	май	доллар	900
Талер	май	евро	500

Задание 2. Сконструировать запросы в БД, созданной вами ранее, согласно варианту из числа ниже перечисленных, в котором говорится о вашей БД.

Вариант 1. В БД **МОДА** сконструировать:

1) запрос на вывод сведений:

Модель	Цена, тыс. р.	Количество отгруженных изделий, шт.
--------	------------------	--

2) запрос на вывод сведений о торговых домах, которым поставлена модель К-9;

3) запрос, позволяющий значения поля «Цена, тыс. р.» увеличить на 3%.

Вариант 2. В БД **АВТОКОМБИНАТ** сконструировать:

1) запрос на вывод сведений:

Ф.И.О. водителя	Номер машины	Дата перевозки
--------------------	-----------------	-------------------

2) запрос на вывод сведений о машинах, у которых «Расход бензина по норме на 100 км, л.» больше 10 л.;

3) запрос, позволяющий значения поля «Расход бензина по норме на 100 км, л.» уменьшить на 1%.

Вариант 3. В БД **ЖЭС** сконструировать:

1) запрос на вывод из БД сведений:

Ф.И.О. квартиросъемщика	Номер квартиры	Количество жильцов
----------------------------	-------------------	-----------------------

2) запрос на вывод сведений о квартиросъемщиках, у которых «Количество жильцов» больше 2-х человек;

3) запрос, позволяющий значения поля «Оплата за газ на 1 человека, р.» увеличить на 2%.

Вариант 4. В БД **ФИНАНСЫ** сконструировать:

1) запрос на вывод из БД сведений:

Клуб	Адрес	Вид услуги
------	-------	------------

2) запрос на вывод сведений о клубах, у которых «Доходы, млн. р.» больше 80;

3) запрос, позволяющий значения поля «Расходы, млн. р.» увеличить на 1%.

Вариант 5. В БД **ВАЛЮТА** сконструировать:

1) запрос на вывод из БД сведений:

Банк	Адрес	Вид валюты
------	-------	------------

2) запрос на вывод сведений о банках, совершающих операции с долларами;

3) запрос, позволяющий значения поля «Продано на сумму, млн. р.» увеличить на 2%.

Задание 3. Сконструировать формы в БД, созданной вами ранее, согласно варианту из числа ниже перечисленных, в котором говорится о вашей БД.

Вариант 1. В БД **МОДА** сконструировать:

1) форму ОТГРУЗКА ИЗДЕЛИЙ для просмотра записей таблицы ОТГРУЗКА;

2) составную форму ИЗДЕЛИЯ, в которой главная форма должна быть на основе таблицы МОДЕЛЬ, а подчиненная – таблицы ОТГРУЗКА.

Вариант 2. В БД **АВТОКОМБИНАТ** сконструировать:

1) форму ПЕРЕВОЗКА ГРУЗОВ для просмотра записей таблицы ПЕРЕВОЗКА;

2) составную форму РЕЙСЫ МАШИН, в которой главная форма должна быть на основе таблицы ВОДИТЕЛЬ, а подчиненная – таблицы ПЕРЕВОЗКА.

Вариант 3. В БД **ЖЭС** сконструировать:

1) форму СВЕДЕНИЯ О КВАРТИРОСЪЕМЩИКАХ для просмотра записей таблицы КВАРТИРОСЪЕМЩИК;

- 2) составную форму ОПЛАТА ЗА ГАЗ, в которой главная форма должна быть на основе таблицы ОПЛАТА, а подчиненная – таблицы КВАРТИРОСЪЕМЩИК.

Вариант 4. В БД **ФИНАНСЫ** сконструировать:

- 1) форму ФИНАНСОВОЕ СОСТОЯНИЕ для просмотра записей таблицы СОСТОЯНИЕ;
 2) составную форму ДЕЯТЕЛЬНОСТЬ КЛУБОВ, в которой главная форма должна быть на основе таблицы КЛУБ, а подчиненная – таблицы СОСТОЯНИЕ.

Вариант 5. В БД **ВАЛЮТА** сконструировать:

- 1) форму ПРОДАЖА ВАЛЮТЫ для просмотра записей таблицы ПРОДАЖА;
 2) составную форму ОПЕРАЦИИ С ВАЛЮТОЙ, в которой главная форма должна быть на основе таблицы БАНК, а подчиненная – таблицы ПРОДАЖА.

Задание 4. Сконструировать отчет в БД, созданной вами ранее, согласно варианту из числа ниже перечисленных, в котором говорится о вашей БД.

Вариант 1. В БД **МОДА** сконструировать отчет вида:

Ведомость отгрузки изделий

Наименование торгового дома	Модель	Количество, шт.	Цена, р.	Сумма, р.
.	.	.	.	?
Итого:				?

Вариант 2. В БД **АВТОКОМБИНАТ** сконструировать отчет вида:

Ведомость расхода бензина

Номер машины	Дата перевозки	Пробег ма- шины, км	Расход бензина по норме на 100 км, л	Всего расход бензи- на по норме, л
.	.	.	.	?
Итого:				?

Вариант 3. В БД **ЖЭС** сконструировать отчет вида:

Ведомость оплаты за газ

Адрес (улица, дом)	Номер квартиры	Количество жильцов	Оплата за газ на 1 человека, р.	Всего за газ, р.
-----------------------	-------------------	-----------------------	------------------------------------	------------------

. . . . ?

Итого: ?

Вариант 4. В БД **ФИНАНСЫ** сконструировать отчет вида:

Чистый доход клубов

Клуб	Вид услуги	Доходы, млн. р.	Расходы, млн. р.	Чистый доход, млн. р.
------	------------	-----------------	------------------	-----------------------

. . . . ?

Итого: ?

Вариант 5. В БД **ВАЛЮТА** сконструировать отчет вида:

Сведения о продаже валюты за месяц

Банк	Вид валюты	Продано на сумму, млн. р.
------	------------	---------------------------

. . .

Итого: ?

Задание 5. Создать файл БД с именем **SQL**.

Задание 6. В БД **SQL** создать с помощью команд языка **SQL** запросы согласно варианту из числа ниже перечисленных.

Вариант 1. Создать:

1) SQL-запросы, реализующие создание таблицы вида:

МОДЕЛЬ	
Модель	Цена, тыс. р.
C/7	250
K-9	300

2) SQL-запрос на выборку из таблицы **МОДЕЛЬ** сведений о модели **C/7**.

Вариант 2. Создать:

1) SQL-запросы, реализующие создание таблицы вида:

ВОДИТЕЛЬ

Ф.И.О. водителя	Номер машины	Расход бензина по норме на 100 км, л
-----------------	--------------	--------------------------------------

Смаль С.С.	52АМ-7	10
Гринь Г.Г.	11МР-6	12

2) SQL-запрос на выборку из таблицы ВОДИТЕЛЬ сведений о водителе *Гринь Г.Г.*

Вариант 3. Создать:

1) SQL-запросы, реализующие создание таблицы вида:

Адрес (улица, дом)	Оплата за газ на 1 человека, р.
Брыля, 20	5000
Сапеги, 11	6000

2) SQL-запрос на выборку из таблицы ОПЛАТА сведений об оплате за газ в доме по адресу: *Брыля, 20.*

Вариант 4. Создать:

1) SQL-запросы, реализующие создание таблицы вида:

Клуб	Адрес
Юность	ул. Садовая, 11
Досуг	ул. Минина, 11

2) SQL-запрос на выборку из таблицы КЛУБ сведений о клубе *Юность.*

Вариант 5. Создать:

1) SQL-запросы, реализующие создание таблицы вида:

Банк	Адрес
Олимп	ул. Победы, 15
Талер	ул. Лесная, 27

2) SQL-запрос на выборку из таблицы БАНК сведений о банке *Олимп.*

ЛИТЕРАТУРА

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник для вузов. СПб.: Питер, 2001.
2. Маклаков С.В. ВРwin и ERwin. Case-средства разработки информационных систем. М.: Диалог-МИФИ, 2008.
3. Муромцев Д.И. Онтологический инжиниринг знаний в системе Protege. СПб.: Спб ГУ ИТМО, 2007.
4. Оскерко В.С., Пунчик З.В. Компьютерные информационные технологии. Часть 2. Базы данных и знаний: Учеб. пособие. Мн.: БГЭУ, 2011.
5. Оскерко В.С., Пунчик З.В., Сосновский О.А. Технологии баз данных. Учеб. пособие. Мн.: БГЭУ, 2007.
6. Сеннов А. Access 2010. Учебный курс.– СПб.: Питер, 2010.
7. Хомоненко А.Д., Мальцев М.Г., Цыганков В.М. Базы данных. Учебник для вузов. 6-е изд.– СПб.: КОРОНА принт, 2009.