

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»
Институт профессионального образования
Кафедра информационных и автоматизированных производственных систем

Составитель О.С. Семенова

ОП. 08 Основы проектирования баз данных

Методические указания к самостоятельной работе

Рекомендовано цикловой методической комиссией
информационных систем и программирования
в качестве электронного издания для использования
в образовательном процессе

Кемерово
2018

Рецензенты:

И.В. Чичерин, к.т.н., доцент кафедры информационных и автоматизированных производственных систем

Семенова Ольга Сергеевна

Основы проектирования баз данных: методические указания к самостоятельной работе [Электронный ресурс]: для студентов специальности СПО 09.02.07 «Информационные системы и программирование» / сост. О.С. Семенова, КузГТУ. – Электрон. издан. – Кемерово, 2018.

Приведенные методические указания к самостоятельной работе по курсу «Основы проектирования баз данных» позволяют углубить знания, полученные в ходе аудиторных занятий; способствуют закреплению теоретических положений; развивают навыки по их практическому применению.

© КузГТУ

© Семенова О.С.,

Составление, 2018

Часть №1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКЕ К ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА

Раздел 1. Основные понятия баз данных

Тема 1.1. Основные понятия теории БД

Содержание темы

1. Основные понятия теории БД
2. Технологии работы с БД

Литература [4].

Методические рекомендации

В процессе изучения данной темы обучающийся основные понятия баз данных.

База данных – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Банк данных – это система специальным образом организованных данных – баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и многоцелевого использования данных.

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

Информационная система – это совокупность тем или иным способом структурированных данных (баз данных) и комплекса аппаратно-программных средств для хранения данных и манипулирования ими.

Предметная область – часть реального мира, для которой необходимо построить информационную модель. Примерами предметной области являются: предприятие, склад, универмаг, банк, университет и т.д.

Объект (сущность) – элемент информационной системы, это нечто существующее и различимое, т.е. объектом можно назвать, то «нечто», для которого существуют название и способ отличать один подобный объект от другого. Объект может быть реальным (человек, предмет, населенный пункт) и абстрактным (событие, счет покупателя, дисциплина).

Классом объектов называется совокупность объектов, обладающих одинаковым набором свойств. Объекты и их свойства являются понятиями реального мира.

Атрибут (свойство) – информационное отображение свойств объекта. Пример: клиент магазина, продающий автомобили имеет такие атрибуты как фамилию, имя, отчество, адрес, и возможно идентификационный номер. Атрибут при реализации информационной модели, на каком либо носителе информации часто называют элементом данных, полем данных или просто полем. Атрибут некоторого набора объектов сам может быть набором объектов, имеющим собственные атрибуты.

Таблица – некоторая регулярная структура, состоящая из конечного набора однотипных записей. Значения данных представляют собой действительные данные, содержащиеся в каждом элементе данных. От того, как элементы данных описывают объект, их значения могут быть количественными, качественными или описательными.

Экземпляр объекта – один набор значений его элементов данных. Запись данных - это совокупность значений связанных элементов данных.

Связь – это функциональная зависимость между сущностями.

Доменом называется набор записей одного типа, отвечающих поставленным условиям.

Ключевым элементом данных называется такой элемент, по которому можно определять значения других элементов данных. Первичный ключ – это атрибут (или группа атрибутов), которые единственным образом идентифицирует каждую строку в таблице. Альтернативный ключ – это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр объекта.

Словарь данных – это централизованное хранилище сведений об объектах, элементах данных, взаимосвязях между объектами, их источниках, значениях, использовании и формах представления данных.

Нормализация отношений – это процесс построения оптимальной структуры таблиц и связей в реляционной модели данных.

SQL (Structured Query Language) – Структурированный Язык Запросов – стандартный язык запросов по работе с реляционными БД.

Триггеры – это программа действий, которая автоматически осуществляется при выполнении операций обновления, добавления или удаления данных. Ни пользователь, ни приложение не могут активизировать триггер. Триггер автоматически выполняется, когда пользователь или приложение выполняют с БД определенные действия. Триггер является мощным инструментом контроля над изменением данных в БД, а также помо-

гает программисту автоматизировать операции, которые должны выполняться в этом случае.

Тема 1.2. Взаимосвязи в моделях и реляционный подход к построению моделей

Содержание темы

1. Логическая и физическая независимость данных
2. Типы моделей данных. Реляционная модель данных
3. Реляционная алгебра

Литература [1,2,4].

Методические рекомендации

В процессе изучения данной темы обучающийся должен усвоить, что база данных (БД) – это совокупность специальным образом организованных и взаимосвязанных данных по конкретной предметной области, хранимых на внешних носителях информации и управляемых средствами СУБД.

В базе данных обеспечивается логическая взаимосвязь хранимых данных и их минимально необходимая избыточность.

По способу организации данных различают:

- Иерархические (рисунок 1)

Иерархическая модель данных – представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами (в программировании применительно к структуре данных дерево устоялось название братья).

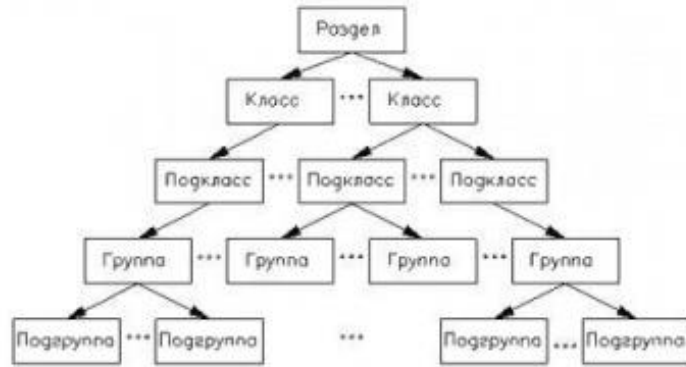


Рисунок 1 – Иерархическая БД

- Сетевые (рисунок 2)

Сетевая СУБД – СУБД, построенная на основе сетевой модели данных. К основным понятиям сетевой модели базы данных относятся: уровень, элемент (узел), связь.

Узел – это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. В сетевой структуре каждый элемент может быть связан с любым другим элементом.

Сетевые базы данных подобны иерархическим, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.

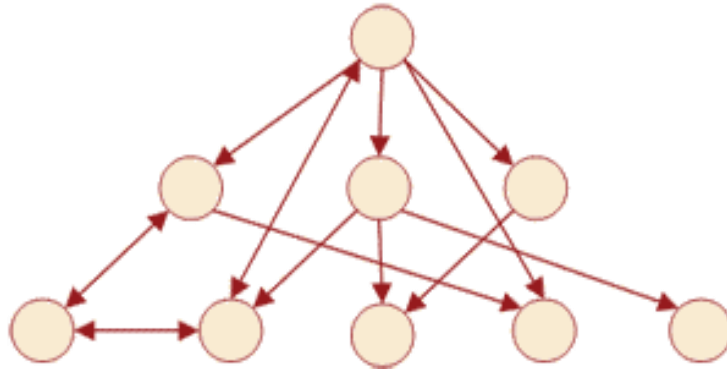


Рисунок 2 – Сетевая БД

- Реляционные базы данных (рисунок 3)

Данные структурированы в виде отдельных таблиц (рисунок 4). Таблицы могут быть связаны между собой.

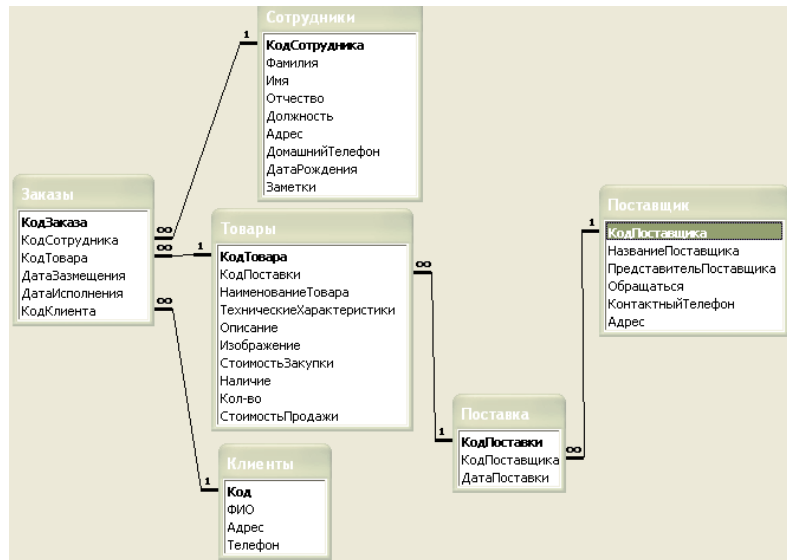


Рисунок – 3 Структура реляционной БД

Маршруты		
ID	Номер маршрута	Тип маршрута
11	9	автобус
12	10	автобус
13	10т	автобус
14	11т	автобус
15	12	автобус
16	13т	автобус
17	14	автобус
18	14т	автобус
19	15т	автобус
20	16	автобус
21	16т	автобус
22	17	автобус
23	17т	автобус
24	18	автобус
25	18т	автобус
26	19	автобус
27	19т	автобус

Рисунок 4 – Таблица реляционной БД

Процесс создания базы данных можно представить в виде трех этапов:

1. Инфологическое (концептуальное) описание баз данных

Исходными данными для осуществления инфологического проектирования является словесная и документальная характеристика предметной области. На этом этапе решается вопрос о том, какие данные должны храниться в базе и какого типа информационные выборки и отчеты могут потребоваться пользователю БД.

2. Логическое проектирование баз данных

На этом этапе осуществляется выбор подходящей системы управления базами данных (СУБД) и представление инфологической модели предметной области в форме структуры базы данных конкретной СУБД. Для реляционных баз данных на этом этапе производится описание структуры каждой таблицы и их взаимосвязей.

3. Физическое проектирование. Предполагает определение способов и мест размещения базы данных, оценку ее объема и других параметров.

Контрольные вопросы:

1. Что такое базы данных?
2. Опишите первые модели данных.
3. Опишите иерархические СУБД.
4. Опишите сетевые базы данных.
5. Что такое реляционная модель данных?
6. Что такое инфологическое проектирование базы данных?
7. Что такое логическое проектирование базы данных?
8. Что такое физическое проектирование базы данных?

Раздел 2. Проектирование баз данных

Тема 2.1. Этапы проектирования баз данных

Содержание темы

1. Основные этапы проектирования БД
2. Концептуальное проектирование БД
3. Нормализация БД.

Литература [1,2,4].

Методические рекомендации

Большинство современных систем управления базами данных (СУБД) разработаны на основе реляционной алгебры.

Первая работа по реляционной модели данных «A Relational Model of Data for Large Shared Data Banks» была опубликована в 1970 г. Её автор - Эдгар Франк Кодд. В своей статье Э. Кодд вывел несколько правил, или форм, по упорядочиванию данных и их отношений.

Нормализация БД – это проектирование базы данных так, чтобы она была компактной и не несла логическую избыточность. Существует несколько разновидностей нормализации, так называемые **нормальные формы**. Все они идут в порядке усложнения от простого к сложному.

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

Всего существует 6 нормальных форм. На практике редко нормализуют выше 3-ей нормальной формы.

Существует специальная терминология, принятая в теории реляционных БД (рисунок 5):

- *Атрибут* соответствует столбцу таблицы, а именно – свойствам объектов, сведения о которых хранятся в ней. В СУБД ACCESS атрибуты называют *полями*.
- *Кортеж* соответствует заполненной строке таблицы. В СУБД ACCESS кортежи называют *записями*.
- *Кардинальное число* – количество кортежей в таблице в текущий момент времени.
- *Домен* – это общая совокупность значений, из которой берутся конкретные значения для конкретного атрибута.

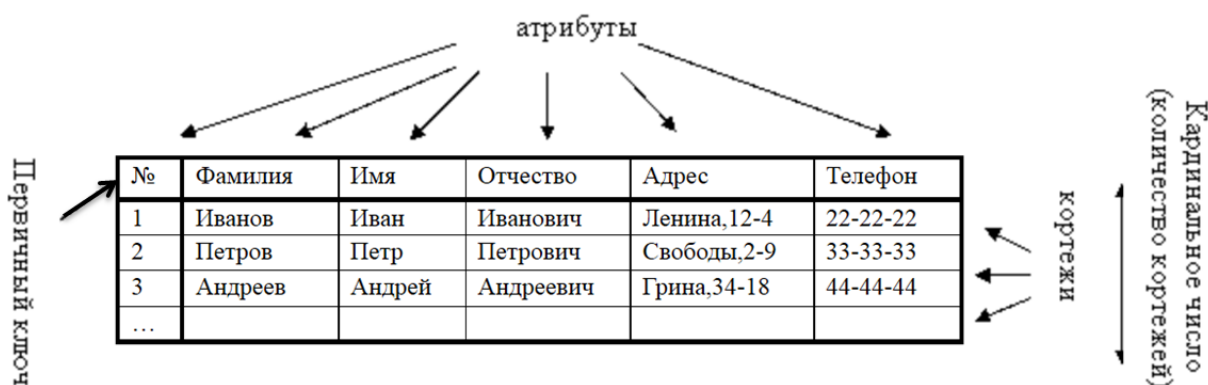


Рисунок 5 – Основные понятия БД

Ключи являются составляющей частью нормализованных таблиц. Бывают двух видов — внешние и первичные.

Первичный ключ — это атрибут, значения которого уникально идентифицируют каждую запись таблицы. Первичный ключ отвечает следующим условиям: он должен иметь значение, не NULL; быть неизменным; иметь уникальное значение для каждой строки.

Внешние ключи – это ссылки на первичные ключи других таблиц.

Отношения – это указатели, которые показывают, как соотносятся данные в одной таблице с данными в другой. Отношения бывают 3-х видов:

- 1) Связь «один-к-одному». Такая связь означает, что каждому значению реквизита *A* соответствует одно и только одно значение связанного с ним реквизита *B*, и наоборот.
- 2) Связь «один-ко-многим». Эта связь означает, что каждому значению реквизита *A* соответствует ноль, одно или несколько значений связанного с ним реквизита *B*, а каждому значению реквизита *B* соответствует одно и только одно значение реквизита *A*.
- 3) Связь «многие-ко-многим». Такая связь означает, что каждому значению

нию реквизита A соответствует несколько значений связанного с ним реквизита B , и наоборот.

Первая нормальная форма (1NF)

Объект базы данных находится в первой нормальной форме тогда, когда каждый ее атрибут *атомарен*. Атрибут атомарен тогда, когда его значение теряет смысл при перестановке любой из его частей или при любом разбиении его на части. То есть, одно поле – одно значение.

Вторая нормальная форма (2NF)

Объект базы данных находится во второй нормальной форме тогда, когда он находится в первой нормальной форме и при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полно зависит от каждого потенциального ключа. Это правило говорит об отделении функционально полных зависимостей на отдельные структуры.

Третья нормальная форма (3NF)

Объект базы данных находится в третьей нормальной форме тогда, когда он находится во второй нормальной форме и отсутствуют транзитивные зависимости не ключевых объектов от ключевых.

Транзитивная зависимость – это очевидная зависимость между полями. Если поле A равно x , то поле B обязательно будет равно y . А если поле B равно z , то тогда поле C будет равно t . Такой зависимости между объектами быть не должно.

Тема 2.2. Проектирование структур баз данных

Содержание темы

1. Средства проектирования структур БД
2. Организация интерфейса с пользователем

Литература [1,2,4].

Методические рекомендации

Основными объектами базы данных являются таблицы, запросы, формы, отчеты, макросы и модули.

Таблица – фундаментальная структура системы управления реляционными базами данных. В Microsoft Access таблица – это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). При этом каждое поле содержит отдельную часть записи (например, фамилию, должность или инвентарный номер). Обычно каждая таблица исполь-

зуется для хранения сведений по одному конкретному вопросу (например, о сотрудниках или заказах).

Запрос – это запрограммированное на специальном языке (SQL) требование к системе на выполнение некоторых действий с записями одной или нескольких таблиц. Запросы создаются пользователем для выборки нужных сведений из одной или нескольких связанных таблиц. С помощью запроса можно также обновить, удалить или добавить данные в таблицы или создать новые таблицы на основе уже существующих.

Форма – это объект Microsoft Access, в котором можно разместить элементы управления, предназначенные для ввода, изображения и изменения данных в полях таблиц.

Отчет – это объект Microsoft Access, который позволяет представлять определенную пользователем информацию в определенном виде, просматривать и распечатывать ее.

Макрос – одна или несколько макрокоманд, которые можно использовать для автоматизации конкретной задачи.

Модуль – программа на VBA, позволяющая автоматизировать сложные операции, которые нельзя описать макросами.

Создание таблиц в СУБД Access возможно в следующих режимах:

- ***Режим таблицы*** (рисунок 6)
- В новой версии Access появилась возможность создавать таблицу, не задумываясь о формате данных, которые вы вводите в соответствующий столбец. Программа автоматически отследит вводимую информацию и предложит соответствующий тип данных и наиболее часто используемый формат представления информации. После ввода текста в ячейку второго столбца с правой стороны появится третий столбец с наименованием «Добавить поле». Для удобства работы с таблицей столбцы необходимо переименовывать, для этого существует контекстное меню.
- ***Режим конструктора*** (рисунок 7)

При создании таблицы в режиме конструктора необходимо самостоятельно прописывать не только названия полей, но и тип данных (таблица 1). В окне конструктора таблиц также устанавливаются необходимые свойства полей (Размер поля, Формат поля, Маска ввода, Пустые строки, Сжатие Юникод, Подпись, Значение по умолчанию, Условие на значение, Сообщение об ошибке, Обязательное поле, Индексированное поле).

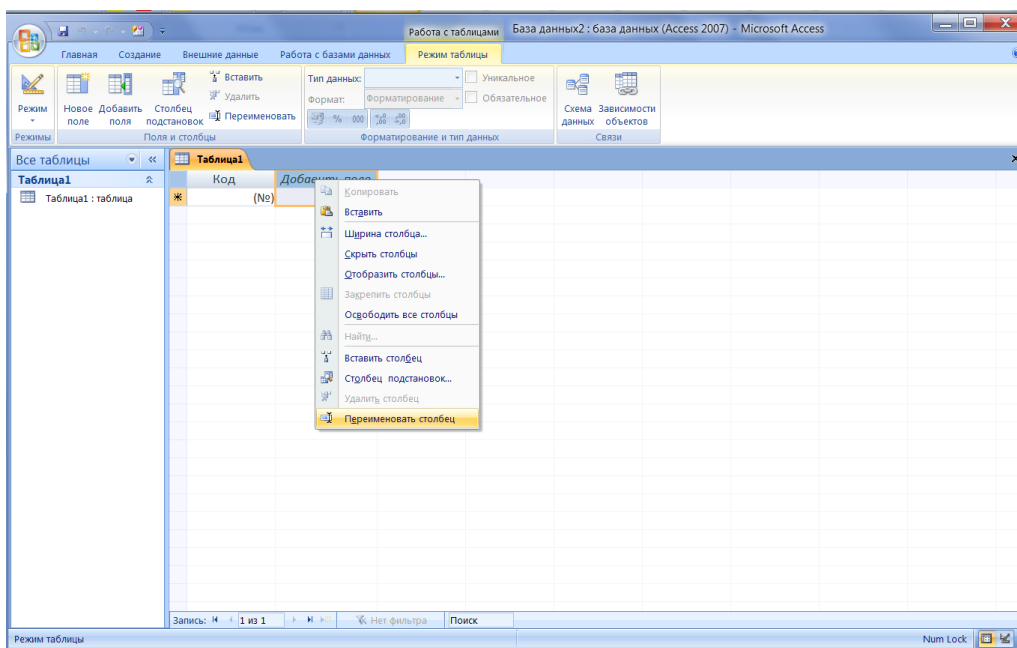


Рисунок 6 – Режим конструктора

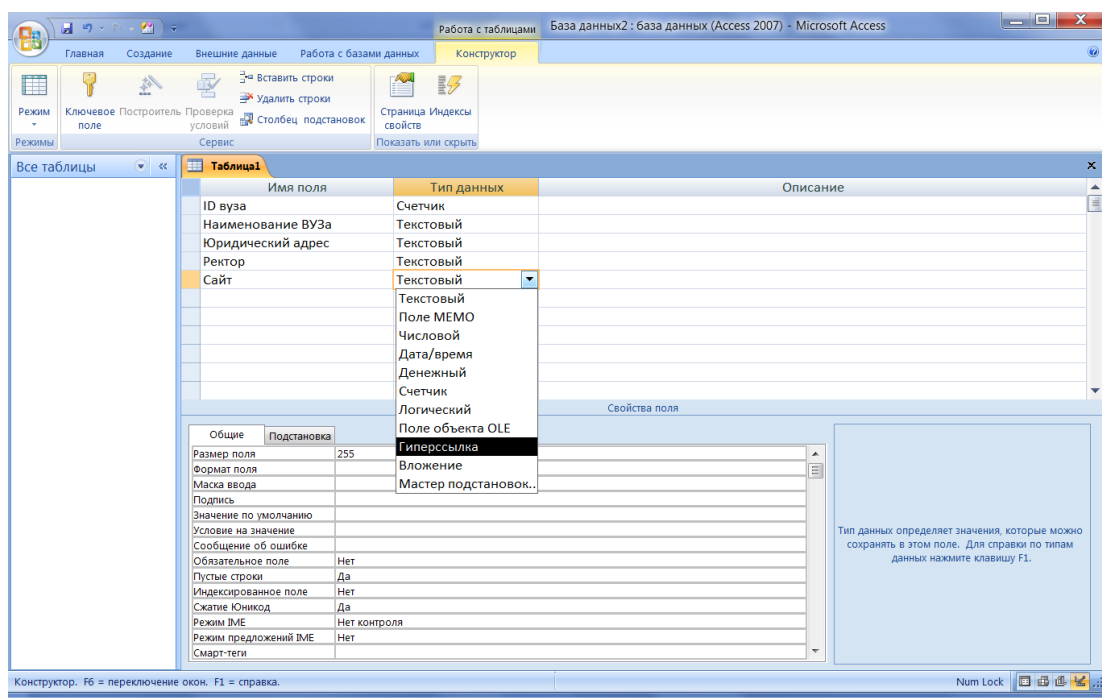


Рисунок 7 – Режим конструктора

Таблица 1 – Типы данных

Тип данных	Описание	Размер
Текстовый	Алфавитно-цифровые символы	0 – 255 символов
Поле MEMO	Алфавитно-цифровые символы, текст большого размера	0 – 64000 символов

Числовой	Числовые значения	1, 2, 4 или 8 байт
Дата/время	Дата и время	8 байт
Денежный	Денежные значения	8 байт
Счетчик	Автоматические номера. Могут быть последовательными, могут - случайными	4 байта
Логический	Логические значения Да/Нет	1 бит (0 или -1)
Поле объекта ОЛЕ	Рисунки, диаграммы, звук и видео	До 1 Гбайт
Гиперссылка	Связь с ресурсом Internet	0 – 6144
Мастер подстановок	Данные, подставляемые из другой таблицы	Обычно 4 байта

Установка связей между таблицами

Это последний этап проектирования БД. На этом этапе *фактически* регистрируются связи между первичными и внешними ключами. Связи между таблицами устанавливаются на ленте «Работа с базами данных» – «Схема данных» (рисунок 8).

Ошибки при связывании полей возникают если:

- Связываемые поля имеют различный тип данных (Исключение: поле типа счетчик всегда связывается с числовым).
- Данные в полях противоречат друг другу (Внешний ключ содержит данные, отличные от значений первичного ключа).

Организация интерфейса с пользователем осуществляется с помощью форм и отчетов.

Формы предназначены для ввода и просмотра взаимосвязанных данных на экране в удобном виде, который соответствует привычному для пользователя документу (рисунок 9). Формы можно применять для создания панелей управления в приложениях (т. е. добавлять на них кнопки, переключатели, рисунки и т.д.).

Формы можно рассматривать как окна, через которые пользователи могут просматривать и изменять базу данных. Рационально построенная форма ускоряет работу с базой данных, поскольку пользователям не требуется искать то, что им нужно. Внешне привлекательная форма делает работу с базой данных более приятной и эффективной, кроме того, она может помочь в предотвращении неверного ввода данных. В Microsoft Office Access 2007, 2010 предусмотрены новые средства, помогающие

быстро создавать формы, а также новые типы форм и функциональные возможности, благодаря которым база данных становится более практичной.

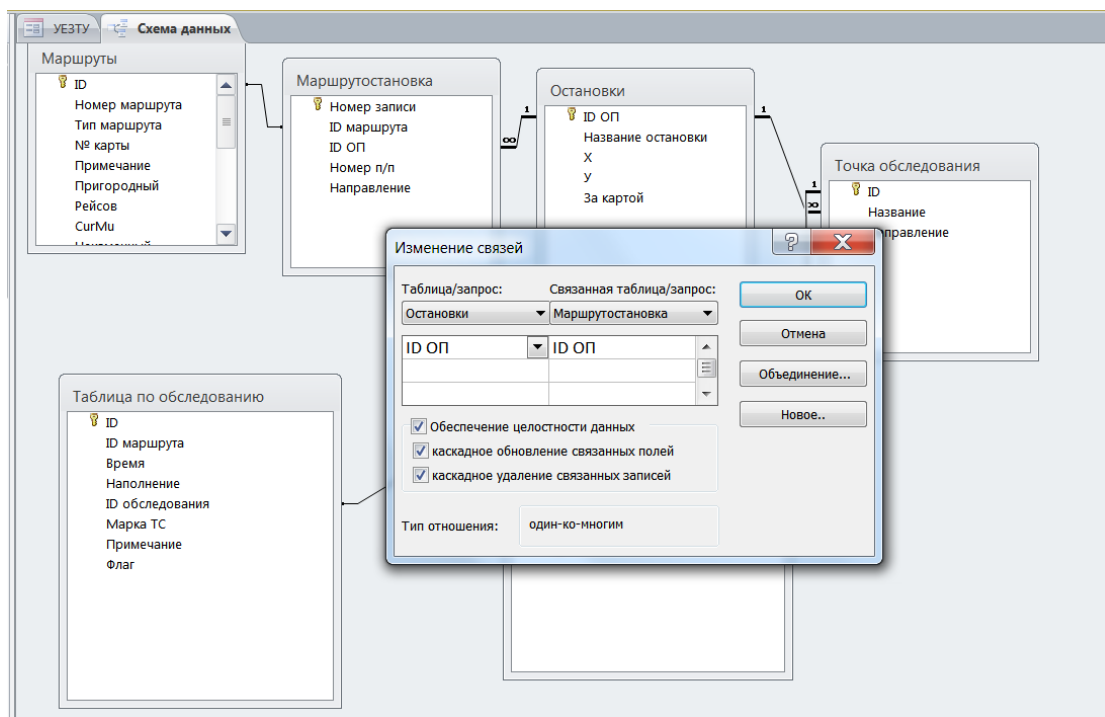


Рисунок 8 – Установка связей между таблицами

При помощи инструмента «Форма» можно создать форму одним щелчком мыши. При использовании этого средства все поля базового источника данных размещаются в форме. Можно сразу же начать использование новой формы либо при необходимости изменить ее в режиме макета или конструктора.

Разделенная форма – это новая возможность в Microsoft Office Access 2007, позволяющая одновременно отображать данные в двух представлениях – в режиме формы и в режиме таблицы.

Эти два представления связаны с одним и тем же источником данных и всегда синхронизированы друг с другом. При выделении поля в одной части формы выделяется то же поле в другой части. Данные можно добавлять, изменять или удалять в каждой части формы (при условии, что источник записей допускает обновление, а параметры формы не запрещают такие действия).

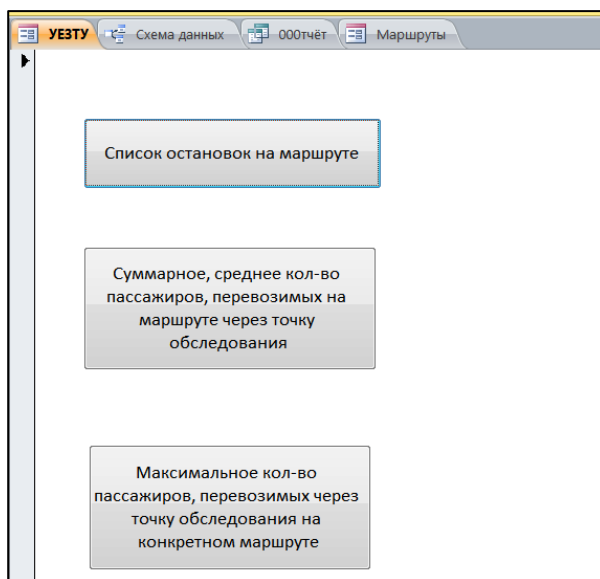


Рисунок 9 – Форма

Работа с разделенной формой дает преимущества обоих типов формы в одной форме. Например, можно воспользоваться табличной частью формы, чтобы быстро найти запись, а затем просмотреть или изменить запись в другой части формы.

Чтобы создать разделенную форму при помощи инструмента «Разделенная форма», выполните следующие действия.

1. В области переходов щелкните таблицу или запрос с данными, которые должны отображаться в форме, или откройте таблицу или запрос в режиме таблицы.
2. На вкладке *Создание* в группе *Формы* щелкните *Разделить форму*.

Приложение Access создаст форму и отобразит ее в режиме макета. В режиме макета можно внести изменения в структуру формы при одновременном отображении данных. Например, при необходимости можно настроить размер полей в соответствии с данными.

В форме, созданной с помощью средства «Форма», одновременно отображается только одна запись. Если нужна форма, в которой отображается сразу несколько записей, и при этом требуются более широкие возможности настройки, чем у таблицы, можно воспользоваться инструментом «Несколько элементов».

1. В области переходов щелкните таблицу или запрос с данными, которые должны отображаться в форме.
2. На вкладке *Создание* в группе *Формы* щелкните *Несколько элементов*.

Для получения большей свободы выбора полей, отображаемых на форме, вместо упомянутых выше инструментов можно воспользоваться мастером форм. Кроме того, можно указать способ группировки и сортировки данных, а также включить в форму поля из нескольких таблиц или запросов, при условии, что заранее заданы отношения между этими таблицами и запросами.

Если мастер или инструменты для создания форм не подходят, можно воспользоваться инструментом «Пустая форма», чтобы создать форму. Таким образом, можно очень быстро построить форму, особенно, если в ней будет лишь несколько полей.

Элементами управления называются улучшающие интерфейс пользователя объекты, которые используются для отображения данных или выполнения других действий и позволяют просматривать данные и работать с ними, например надписи и рисунки. Наиболее часто используемый элемент управления — поле. Кроме того, используются такие элементы управления, как надписи, флажки и элементы управления подчиненных форм и отчетов.

Элементы управления могут быть присоединенными, свободными и вычисляемыми.

- *Присоединенный элемент управления* – элемент управления, источником данных которого служит поле таблицы или запроса, называется присоединенным элементом управления. Присоединенный элемент управления служит для отображения значений полей базы данных. Значения могут быть текстовыми, числовыми, логическими, датами, рисунками или диаграммами. Например, для текстового поля в форме, в котором отображается фамилия служащего, могут использоваться данные поля «Фамилия» в таблице «Служащие».
- *Свободные элементы управления* – элементы управления, не имеющие источника данных (например, поля или выражения). Свободные элементы управления используются для вывода на экран сведений, линий, прямоугольников и рисунков. Примером свободного элемента является надпись, которая отображает заголовок формы.

Вычисляемые элементы управления – элементы управления, источником данных которых является выражение, а не поле. Для задания значения, которое должно содержаться в таком элементе управления, необходимо задать *выражение*, служащее источником данных элемента. Выражение – это сочетание операторов (таких как = и +), имен других

элементов управления, имен полей, функций, возвращающих единственное значение, и констант.

Отчеты используются для формирования выходного документа, предназначенного для вывода на печать. Отчёт удобно создавать с помощью *Мастера отчётов* (рисунок 10).

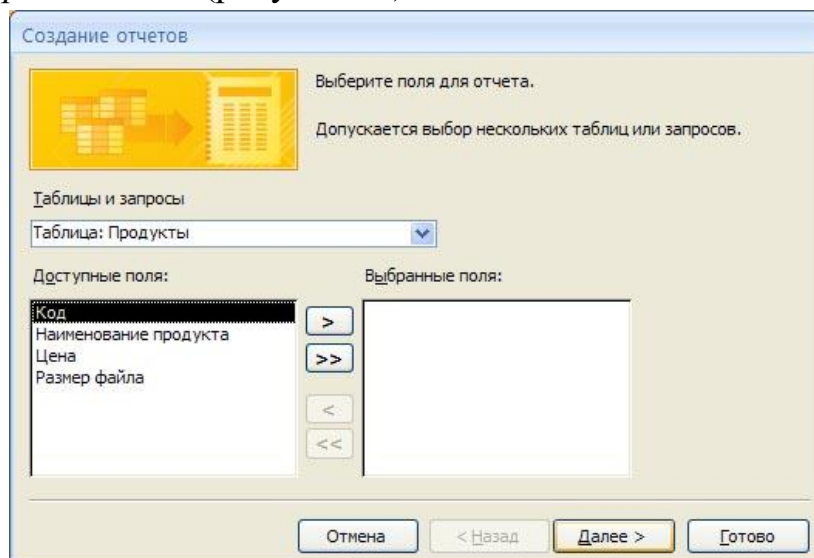


Рисунок 10 – Мастер отчётов

Редактировать отчёт можно в конструкторе отчётов (рисунок 11).

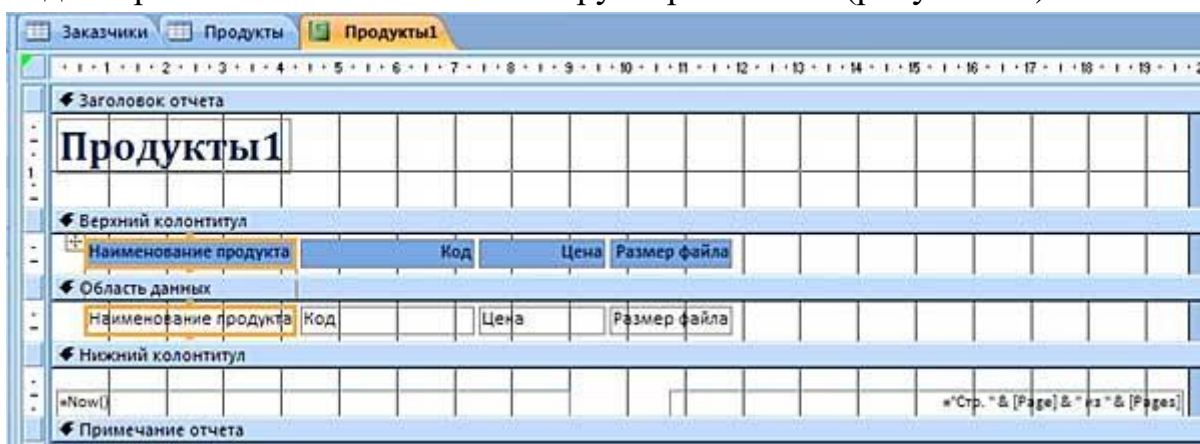


Рисунок 11 – Конструктор отчётов

В режиме просмотра данный отчёт будет выглядеть так, как показано на рисунке 12.

Продукты

Наименование продукта	Цена	Размер файла
Как быстро создать информационный продукт	0,9321	1 000
Как определить будущих потребителей информационного продукта	4,7531	30 450
Стратегия продаж в Интернете	12,3456	1 300

28 сентября 2007г. Стр. 1 из 1

Рисунок 12 – Режим просмотра

Тема 2.3. Организация запросов SQL

Содержание темы

1. Основные понятия языка SQL. Синтаксис операторов, типы данных.
2. Команды языка SQL для определения, обработки и администрирования данных.
3. Организация запросов на выборку данных, обновление, удаление. Сортировка и группировка данных в SQL.

Литература [1,2,4].

Методические рекомендации

Запрос – это запрограммированное на специальном языке (SQL) требование к системе на выполнение некоторых действий с записями одной или нескольких таблиц. Запросы создаются пользователем для выборки нужных сведений из одной или нескольких связанных таблиц. С помощью запроса можно также обновить, удалить или добавить данные в таблицы или создать новые таблицы на основе уже существующих.

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

- создание в базе данных новой таблицы;
- добавление в таблицу новых записей;
- изменение записей;
- удаление записей;
- выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
- изменение структур таблиц.

Со временем SQL усложнился – обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) – и стал приобретать черты, свойственные языкам программирования.

Для создания запросов с помощью языка структурированных запросов необходимо знать, что SQL предназначен для манипулирования данными в реляционных базах данных, определения структуры баз данных и для управления правами доступа к данным в многопользовательской среде. Поэтому, в язык SQL в качестве составных частей входят:

- язык манипулирования данными (Data Manipulation Language, DML)

- язык определения данных (Data Definition Language, DDL)
- язык управления данными (Data Control Language, DCL).

Язык манипулирования данными используется, как это следует из его названия, для манипулирования данными в таблицах баз данных. Он состоит из 4 основных команд:

SELECT (выбрать)
 INSERT (вставить)
 UPDATE (обновить)
 DELETE (удалить)

Язык определения данных используется для создания и изменения структуры базы данных и ее составных частей – таблиц, индексов, представлений (виртуальных таблиц), а также триггеров и сохраненных процедур. Основными его командами являются:

CREATE DATABASE (создать базу данных)
 CREATE TABLE (создать таблицу)
 CREATE VIEW (создать виртуальную таблицу)
 CREATE INDEX (создать индекс)
 CREATE TRIGGER (создать триггер)
 CREATE PROCEDURE (создать сохраненную процедуру)
 ALTER DATABASE (модифицировать базу данных)
 ALTER TABLE (модифицировать таблицу)
 ALTER VIEW (модифицировать виртуальную таблицу)
 ALTER INDEX (модифицировать индекс)
 ALTER TRIGGER (модифицировать триггер)
 ALTER PROCEDURE (модифицировать сохраненную процедуру)
 DROP DATABASE (удалить базу данных)
 DROP TABLE (удалить таблицу)
 DROP VIEW (удалить виртуальную таблицу)
 DROP INDEX (удалить индекс)
 DROP TRIGGER (удалить триггер)
 DROP PROCEDURE (удалить сохраненную процедуру)

Язык управления данными используется для управления правами доступа к данным и выполнением процедур в многопользовательской среде. Более точно его можно назвать "язык управления доступом". Он состоит из двух основных команд:

GRANT (дать права)
 REVOKE (забрать права)

SQL остаётся единственным механизмом связи между прикладным программным обеспечением и базой данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов:

- 1) мастера запросов, который работает в диалоговом режиме;
- 2) конструктора запросов.

Принцип создания запросов с помощью конструктора запросов прост: после запуска конструктора автоматически появляется окно «Добавление таблицы», в котором выбирается одна или несколько таблиц, необходимых для решения поставленных целей; затем устанавливаются связи между таблицами, добавляются необходимые поля, устанавливается порядок сортировки, условия и т.д.

Типы запросов:

1. *Запрос на выборку.* Позволяет выбрать записи из одной или нескольких таблиц согласно условию и представить их в табличной форме.

Для создания запроса необходимо ввести значения следующих параметров (рисунок 13):

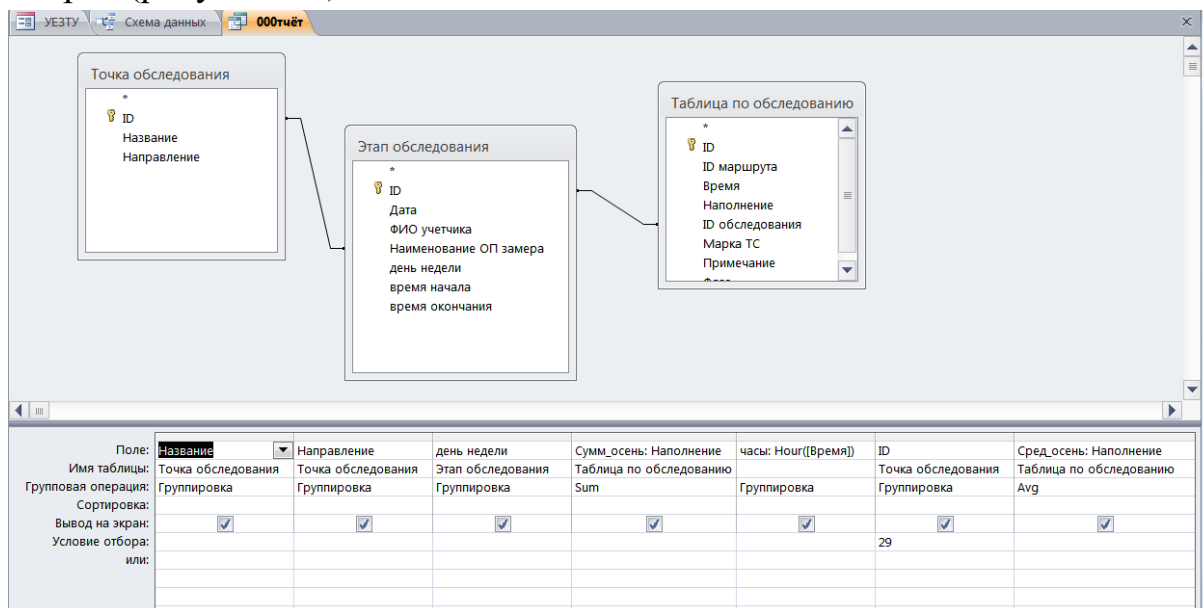


Рисунок 13 – Конструктор запросов. Запрос на выборку

- «Поле» – вводится имя поля
- «Имя таблицы» – вводится имя таблицы
- «Сортировка» – указывается тип сортировки
- «Вывод на экран» – указывается, нужно ли значение поля выводить на экран

- «Условие отбора» – вводится условие для отбора данных из поля.

Для выполнения запроса необходимо на ленте выбрать команду «Выполнить». Результат выводится в виде таблицы (рисунок 14).

часы	Сумм_осень	Сред_осень	Название	Направление	день недели	ID
5	8	1,14285714285714	Строммаш	из города	суббота	29
7	117	10,6363636363636	Строммаш	из города	суббота	29
8	198	12,375	Строммаш	из города	суббота	29
9	157	10,4666666666667	Строммаш	из города	суббота	29
10	151	9,4375	Строммаш	из города	суббота	29
11	131	8,1875	Строммаш	из города	суббота	29
12	160	11,4285714285714	Строммаш	из города	суббота	29
13	204	14,5714285714286	Строммаш	из города	суббота	29
14	212	13,25	Строммаш	из города	суббота	29
15	230	13,5294117647059	Строммаш	из города	суббота	29
16	215	14,3333333333333	Строммаш	из города	суббота	29
17	238	11,9	Строммаш	из города	суббота	29
18	246	20,5	Строммаш	из города	суббота	29
19	279	19,9285714285714	Строммаш	из города	суббота	29
20	98	14	Строммаш	из города	суббота	29
6	40	3,63636363636364	Строммаш	из города	четверг	29
7	269	16,8125	Строммаш	из города	четверг	29
8	236	11,2380952380952	Строммаш	из города	четверг	29
9	174	7,25	Строммаш	из города	четверг	29
10	136	5,44	Строммаш	из города	четверг	29
11	162	7,04347826086957	Строммаш	из города	четверг	29
12	276	14,5263157894737	Строммаш	из города	четверг	29
13	224	11,2	Строммаш	из города	четверг	29
14	207	10,8947368421053	Строммаш	из города	четверг	29
15	272	14,3157894736842	Строммаш	из города	четверг	29
16	354	15,3913043478261	Строммаш	из города	четверг	29
17	552	25,0909090909091	Строммаш	из города	четверг	29
18	545	20,9615384615385	Строммаш	из города	четверг	29
19	401	17,4347826086957	Строммаш	из города	четверг	29

Рисунок 14 – Выборка информации из таблиц

2. *Запрос на добавление.* Добавление записей в таблицу из текущей или внешней базы данных.
3. *Запрос на обновление.* Позволяет изменить записи в одной или нескольких таблицах согласно условию.
4. *Запрос на удаление.* Позволяет удалить записи из таблиц согласно условию.
5. *Запрос на создание таблицы.* Создание таблицы из внешнего файла БД или из таблицы в текущей БД.
6. *Перекрестный запрос.* Позволяет вывести данные из таблиц в компактной форме.

Тема 2.4. Управление объектами базы данных.

Содержание темы

1. Создание макросов и модулей
2. Использование языка программирования VBA для обращения к объектам БД

Литература [3].

Методические рекомендации

Макросы содержат описание действий, которые должны быть выполнены в ответ на некоторое событие. Каждое действие реализуется макрокомандой. Макрос позволяет объединить разрозненные операции обработки данных в одном приложении.

Макрос представляет собой набор макрокоманд, т.е. команд по управлению элементами базы данных (таблицами, формами, запросами и т.д.), сохраненных под одним общим именем. Выполнение макроса – это последовательное выполнение команд, составляющих его. Макросы обычно предназначены для автоматизации несложных, но часто выполняемых действий.

Большинство макрокоманд, из которых составляются макросы, имеют аргументы. Например, макрокоманда *Открыть таблицу* имеет аргумент *Имя таблицы* (т.е. имя таблицы, которую требуется открыть), а также некоторые другие аргументы, задающие режим открытия таблицы. Некоторые макрокоманды не имеют аргументов (например, макрокоманда *Свернуть* просто свертывает текущее окно). Имеется также специальная макрокоманда *Выполнить команду*; ее аргументом является название одного из элементов меню.

Окно описания макроса (вызываемое кнопкой *Создать* на вкладке *Макросы*) содержит две основные колонки: *Макрокоманда* (в этой колонке вводятся макрокоманды, т.е. фактически описывается макрос) и *Примечание* (комментарий). Если макрос должен выполняться при определенном условии, то используется также дополнительная колонка *Условие*.

Макросы могут запускаться на выполнение непосредственно (из вкладки *Макросы*), а также подключаться к кнопкам (обычно размещаемым в окнах форм) или выполняться при определенных условиях (например, при вводе определенных данных).

Рассмотрим пример создания макроса. Одно из самых простых и распространенных применений макросов – одновременное открытие нескольких элементов базы данных (таблиц, форм и т.д.) в желаемом виде. Пусть требуется разработать макрос, который будет одновременно открывать таблицу *Объекты*, а также запрос *Стоимость контрактов по заказчикам*, содержащий данные о стоимости и количестве заказов каждого заказчика. Кроме того, при просмотре этих данных требуется исключить возможность внесения изменений в таблицу данных об объектах.

Макрос, реализующий эти действия, создается следующим образом.

1. Перейти на вкладку Макросы. Нажать кнопку Создать.
2. В первой строке колонки Макрокоманды ввести макрокоманду *Открыть таблицу*. Указать аргументы этой макрокоманды: *Имя таблицы – Объекты, Режим – Таблица, Режим данных – Только чтение* (чтобы исключить внесение изменений в таблицу).
3. В следующей строке ввести макрокоманду *Открыть запрос*. Указать ее аргументы: *Имя запроса - Стоимость контрактов по заказчикам, Режим – Таблица, Режим данных – Только чтение*.
4. В следующей строке ввести макрокоманду *Выполнить команду*. В качестве ее аргумента выбрать команду *РядомГоризонтально* (в некоторых версиях – *TileHorizontally*). Это требуется для того, чтобы окна, открытые предыдущими макрокомандами, оставались на экране (друг над другом).
5. Закрыть окно макроса. На запрос о сохранении внесенных изменений ответить *Да*. Указать имя макроса: *Просмотр информации об объектах*.
6. Чтобы выполнить макрос, необходимо выделить его отметку и нажать кнопку *Запуск* (или просто дважды щелкнуть по отметке макроса). Запрос и таблица выводятся на экран.

Полученное представление данных имеет недостаток: на экране (в нижней части) остается главное окно базы данных. Желательно сворачивать его, прежде чем выводить на экран два других окна (таблицу и запрос). Для этого внесем изменения в макрос.

1. Перейти на вкладку *Макросы*. Выделить макрос *Просмотр информации об объектах*. Нажать кнопку *Конструктор*.
2. Поместить курсор в первую строку (т.е. в ту строку, перед которой требуется вставить новую макрокоманду). Выбрать команду *Вставка – Строки*.
3. В появившейся пустой строке (первой) ввести макрокоманду *Свернуть*. Аргументов у этой макрокоманды нет.
4. Сохранить макрос. Закрыть его (выйти из режима конструктора).
5. Выполнить макрос.

Модули содержат программы на языке Visual Basic, которые могут разрабатываться пользователем для реализации нестандартных процедур при создании приложения (рисунок 15).

Существует два типа модулей: *стандартные модули* и *модули класса*. *Стандартные модули* содержат общие процедуры, которые могут ис-

пользоваться при обработке событий в различных объектах, для вычисления значений в формах, отчетах и т.д.

Эти процедуры не связаны с конкретным объектом и могут использоваться другими приложениями Access.

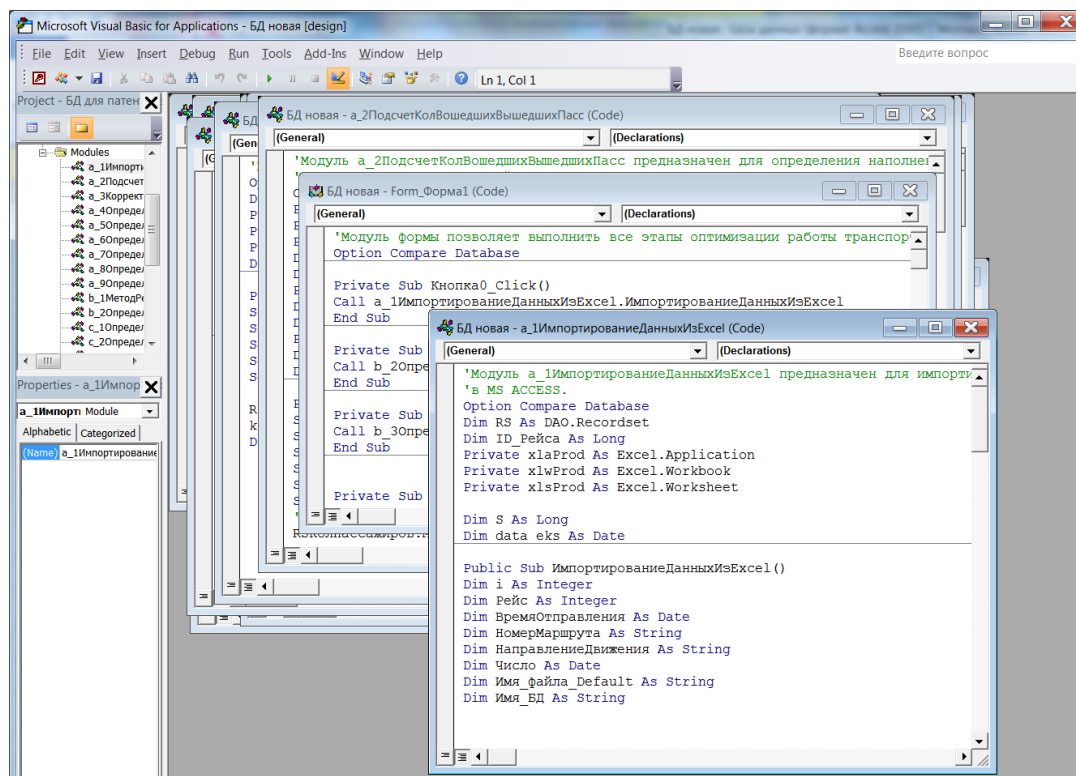


Рисунок 15 – Пример модуля

Вторым типом модуля в Access является *модуль класса*. *Модуль класса* отличается от стандартного модуля тем, что кроме процедур он содержит описание объекта и используется для создания объектов. Процедуры, определенные в этом модуле, являются методами и свойствами объекта. Существует два типа модулей класса: *базовые модули класса*, *модули форм* и *отчетов*. Модули форм и отчетов связаны с соответствующими формами и отчетами и отличаются от базовых модулей лишь тем, что существуют в памяти только до тех пор, пока открыты связанные с ними форма или отчет.

Чтобы создать стандартный модуль или модуль класса, нужно: 1) выбрать команды *Модули* → *Создать*; 2) создать необходимые процедуры на языке программирования Visual Basic for Application, 3) сохранить модуль.

Для работы с данными, размещенными в БД, можно воспользоваться элементом управления данными Data, который осуществляет доступ к данным, используя тот же процессор баз данных Microsoft Jet, что работает

в Microsoft Access. Эта технология обеспечивает прямой доступ ко многим стандартным форматам баз данных, и позволяет писать приложения обработки данных без программирования.

Чтобы создать приложение базы данных, нужно добавить элементы управления данными на форму, точно так же, как это делается с любым другим элементом управления Visual Basic. В форме можно использовать столько элементов управления данными, сколько необходимо. Как правило, для каждой таблицы базы данных, которой необходимо управлять, используется один элемент управления данными.

Чтобы создать простое приложение базы данных необходимо:

1. Добавить элемент управления данными на форму.
2. Установить его свойства (указать базу данных и таблицу, из которых нужно получать данные).
3. Добавить связанные элементы управления (например, окна с текстом, окна списков, и другие элементы управления, которые необходимо связать с элементом управления данными.)
4. Указать в свойствах связанных элементов управления источник данных и поля данных, которые предполагается выводить.

При работе приложения, этот связанный элемент управления автоматически будет показывать поля текущих записей в базе данных.

Как все элементы управления Visual Basic, элементы управления данными имеют ряд свойств, которые надо указать в окне Properties во время разработки. Многие из этих свойств можно установить во время выполнения, в программе на языке Visual Basic.

Элемент управления данными обеспечивает *реляционный (relational)* интерфейс к файлам базы данных. В основе своей реляционная база данных это такая база данных, которая хранит данные таблиц (*tables*), состоящих из *столбцов (columns)* и *строк (rows)*, В Visual Basic, столбцы называются *полями (fields)*, а строки называются *записями (records)*.

Элемент управления данными предлагает высокий уровень функциональности, которым можно воспользоваться, не написав ни строчки программы – просто устанавливая его свойства и управляя ими, и включая в проект связанные с данными элементы управления. Однако для расширения функций элемента управления данными в программе на Visual Basic можно самостоятельно манипулировать элементом управления данными и создаваемым им объектом *Recordset*.

Когда приложение запущено, элемент управления данными работает

вместе с базой данных, предоставляя доступ к текущей совокупности записей, или набору записей (Recordset). То есть, элемент управления данными создает объект Recordset. Как и у любого объекта, у этого объекта есть свойства, которые его характеризуют, и методы (действия, которые объект может совершать). Навигация предполагает передвижение или изменение текущей записи в наборе записей. Навигация осуществляется с помощью методов (таблица 2).

Таблица 2 – Методы объекта Recordset

Метод	Описание	Пример
AddNew	Добавление новой пустой записи к таблице базы данных	Data1.Recordset.AddNew
MoveNext	Перемещение указателя записи на следующую запись в объекте Recordset	Data1.Recordset.MoveNext
MoveFirst	Перемещение указателя записи на первую запись в объекте Recordset	Data1.Recordset.MoveFirst
MoveLast	Перемещение указателя записи на последнюю запись в объекте Recordset	Data1.Recordset.MoveLast
Update	Сохранение данных в новой записи. Добавление новой записи к таблице.	Data1.Recordset.Update
Delete	Удаление текущей записи	Data1.Recordset.Delete
Close	Закрытие набора записей	Data1.Recordset.Close
Edit	Редактирование текущей записи	Data1.Recordset.Edit

Обращение к данным, находящимся в полях, осуществляется с помощью свойства Fields (таблица 3).

Синтаксис, применяемый для управления объектами DAO Database, аналогичен тому, который используется для управления другими объектами Visual Basic. Таким образом, если требуется написать программу перехода к последней записи в наборе записей, можно обратиться к набору записей, как к объекту, а затем применить к нему метод MoveLast:

Data1.Recordset.MoveLast

Если нужно считать значение определенного поля в текущей записи, нужно написать

MyString = Data1.Recordset.Fields("Title").Value

Таблица 3 – Свойства объекта Recordset

Свойство	Описание	Пример
Fields	Значение поля. К полю можно обращаться по имени или по номеру. Поля нумеруются, начиная с нуля.	Data3.Recordset.Fields("Направление").Value = направление ИЛИ ФИО = Data2.Recordset.Fields(2).Value
Eof	Признак окончания таблицы. Если достигнута последняя запись в таблице, то свойство принимает значение True, иначе – False	If Data2.Recordset.EOF = True then Form1.print "Вся таблица просмотрена" End IF

Контрольные вопросы:

1. Что такое базы данных?
2. Опишите первые модели данных.
3. Опишите иерархические СУБД.
4. Опишите сетевые базы данных.
5. Что такое реляционная модель данных?
6. Общая характеристика реляционной модели данных. Типы данных, используемые в реляционной модели.
7. Первичные ключи.
8. Внешние ключи.
9. Операции, которые могут нарушить ссылочную целостность.
10. Стратегии поддержания ссылочной целостности.
11. Этапы разработки базы данных.
12. Первая нормальная форма.
13. Вторая нормальная форма.
14. Третья нормальная форма.
15. Как организовать интерфейс с пользователем?
16. Опишите средства проектирования структур БД.
17. Опишите процесс создания отчета.
18. Опишите процесс создания запросов.

19. Опишите операторы SQL, предназначенные для создания, удаления таблиц.
20. Опишите операторы SQL, предназначенные для выборки данных из таблиц.
21. Опишите операторы SQL, предназначенные для добавления/удаления полей из таблиц.
22. Для чего используется запрос на обновление?
23. Как производится отбор заменяемых записей?
24. Как Access сообщает об обновляемых записях?
25. Как проверить результат выполнения запроса?
26. Что осуществляет запрос на добавление записей?
27. Как можно узнать о числе обновляемых записей?
28. Какое есть условие для успешного выполнения запроса на добавление?
29. Что позволяет выполнить запрос на удаление?
30. Как выполняется запрос на удаление, если установлен параметр *Каскадное удаление связанных записей*?
31. Как выполняется запрос на удаление, если не установлены параметры целостности *Каскадное удаление связанных записей*?
32. Как осуществляется обращение к БД с помощью языка программирования Visual Basic?
33. Опишите принципы использования элемента управления данными. Какие свойства объекта Data требуется настроить для доступа к БД?
34. Что такое набор записей в БД?
35. Как осуществляется навигация по набору записей?
36. Какие свойства и методы объекта RecordSet вы знаете?

Часть №2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ДОМАШНИХ ЗАДАНИЙ

1. Используя иерархическую, сетевую, реляционную, постреляционную, многомерную, объектно-ориентированную модели данных структурировать информацию из заданной предметной области.

Литература: [4]

1. Построить иерархическую, сетевую модель данных. Указать корневой тип, подтипы, узлы.
 - а) Генеалогическое древо;
 - б) Сайт kuzstu.ru;
 - в) Перевозка пассажиров между 5-ю населенными пунктами;
 - г) Транспортировка грузов со склада по 10 пунктам разгрузки;
 - д) Улично-дорожная сеть из 6-и транспортных узлов;
2. Построить реляционную модель данных. Указать атрибуты, кортежи, степень отношения, кардинальное число, домен.
 - а) Школьники;
 - б) Компьютерные комплектующие;
 - в) Запчасти к автомобилю;
3. Построить многомерную модель данных.
 - а) Расписание движения поездов;
 - б) Прибыль организаций за 10 лет;
 - в) Продажи топлива;
4. Построить объектно-ориентированную модель данных.
 - а) Объектная модель Word.
 - б) Объектная модель Excel

2. Разработать структуру базы данных

Литература: [1,2,4]

Разработка структуры БД начинается с изучения метода нормализации и метода построения ER-модели. Используя один из этих методов, строится логическая модель БД, производится разбивка на таблицы.

Затем определяется генеральный список полей. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Цель – обеспечить, чтобы ввод данных в одну таблицу производил

ся, по возможности, в рамках одного подразделения, а еще лучше – на одном рабочем месте.

В каждой из таблиц намечают *ключевое поле*. В качестве такого выбирают поле, данные в котором повторяться не могут. Если в таблице нет полей, которые можно было бы использовать, как ключевые, всегда можно ввести дополнительное поле типа Счетчик – оно не может содержать повторяющихся данных по определению.

Определяют связи между таблицами. Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым.

3. Заполнить таблицы БД информацией. Создать запросы, отчёты и формы.

Литература: [1,2,4]

Используя разработанную логическую модель БД, создать физическую БД в среде Ms Access. Произвести наполнение таблиц БД в соответствии с заданной предметной областью. Создать запросы и отчёты в соответствии с вариантом. Создать формы для вызова запросов, созданных в задании. Создать формы для внесения новой информации в БД.

Варианты домашних заданий Дз1-Дз3:

Вариант 1. База данных Туризм.

- Создать запрос для отображения всей информации о фирмах, предлагающих путевки на отдых в горнолыжных базах.
- Создать запрос для отображения информации о стоимости путевок в Египет с учетом предоставляемых скидок.
- Создать отчет по обеим таблицам.

Вариант 2. База данных Библиотека.

- Создать параметрический запрос для отображения фамилий и телефонов учеников, которые должны сдать книги до даты, определенной параметром, и названий этих книг.
- Создать запрос для отображения числа учеников каждого класса, взявших книги.
- Создать отчет по обеим таблицам.

Вариант 3. База данных Банк.

- Создать запрос на создание таблицы, отображающей информацию о курсах продажи и покупки долларов США и Канады.

- Создать запрос для отображения информации о сумме продажи по всем отделениям.
- Создать отчет по продаже и покупке валюты по всем отделениям с указанием общей суммы.

Вариант 4. База данных Магазин.

- Создать параметрический запрос для отображения всей информации о фирмах, поставляющих определенным значением параметра товар.
- Создать запрос для отображения общей суммы поставок каждого товара.
- Создать отчет по второй таблице, включив все поля, указать общую сумму поставок каждого товара.

Список литературы

1. Федорова, Г. Н. Основы проектирования баз данных [Текст] : учебное пособие для образовательных учреждений, реализующих программы среднего профессионального образования по специальности "Информационные системы (по отраслям)" : [для студентов СПО] / Г. Н. Федорова. – Москва : Академия, 2017. – 224 с. – Доступна электронная версия: <http://www.academia-moscow.ru/catalogue/4831/296505/>
2. Голицына, О. , Л. Основы проектирования баз данных. – Москва : НИЦ ИНФРА-М, 2016. – 416 с. – Режим доступа: <http://znanium.com/go.php?id=552969>. – Загл. с экрана. (14.12.2018)
3. Шустова, Л. И. Базы данных. – Москва : НИЦ ИНФРА-М, 2018. – 304 с. – Режим доступа: <http://znanium.com/go.php?id=967755>. – Загл. с экрана. (14.12.2018)
4. Цветкова, М. С. Информатика [Электронный ресурс] : учебник для использования в учебном процессе образовательных учреждений СПО на базе основного общего образования с получением среднего общего образования / М. С. Цветкова, И. Ю. Хлобыстова. – Москва : Академия, 2017. – 352 с. – Режим доступа: <http://academia-moscow.ru/reader/?id=227485#copy>. – Загл. с экрана. (14.12.2018)

Содержание

Часть №1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКЕ К ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА	3
Часть №2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ДОМАШНИХ ЗАДАНИЙ.....	29
Список литературы.....	32