2. Цели занятия:

- формирование представления о способах организации запросов к базам данных.

- изучение основополагающих принципов разработки и использования современных информационных технологий.

3. Учебные вопросы:

1. Определение элементов запросов к базам данных.
2. Формирование логических выражений для создания запроса.
3. Элементы языка SQL и запросы в форме SQL

Вопрос 1. Определение элементов запросов к базам данных.

Запросы создаются пользователем для выборки необходимых ему данных из одной или нескольких связанных таблиц и представления выбранных данных также в виде таблицы. Запрос может формироваться двумя способами:

- с помощью запросов по образцу — QBE (Query By Example);
- с помощью инструкций языка структурированных запросов SQL (Structured Query Language), т.е. специализированного языка, предназначенного для организации запросов, а также для обновления и управления реляционными базами данных.

Практически все типы запросов в Access можно создать визуально. Исключение составляют сквозные запросы (SQL-pass-through), т.е. запросы из других приложений, запросы на изменение структуры данных и запросы объединения.

Визуально можно построить запросы добавления, удаления, обновления и создания таблиц.

Отметим также, что одной из наиболее сильных сторон Access являются фильтры, которые строятся с помощью запросов или посредством установки критериев. Для облегчения этой задачи используют параметрические запросы.

В Access имеется несколько видов запросов:

- запрос на выборку, т.е. выбирающий данные из взаимосвязанных таблиц и других запросов. В результате получают таблицу, существующую до закрытия запроса. Таблицу с результатами запроса можно использовать для работы с данными таблиц, на которых построен запрос;
- запрос на создание таблицы, основанный на запросе на выборку, но в отличие от последнего результат этого запроса сохраняется в новой таблице;
- запросы на обновление, добавление, удаление, являющиеся запросами действия, в результате выполнения которых изменяются данные в таблицах.

Основные принципы конструирования запроса заложены в технике конструирования запроса на выборку, являющегося основой всех видов запросов.

Запрос на выборку позволяет достаточно просто выбрать данные из одной или нескольких взаимосвязанных таблиц. Результаты запроса отображаются в виде таблицы.

При конструировании запроса достаточно выделить и перетащить необходимые поля из таблиц, представленных в схеме данных запроса, в бланк запроса и

ввести условия отбора записей.
Результаты выполнения запроса выводятся в режиме таблицы. Несмотря на то, что поля результирующей таблицы принадлежат, как правило, нескольким таблицам базы данных, с ними можно работать так, как если бы они принадлежали одной таблице. Можно также менять данные в таблице результатов запроса на выборку, при этом сделанные изменения будут внесены в базовые таблицы.

Для создания запроса в окне базы данных надо выбрать закладку Запрос и нажать кнопку [Создать]. В открывшемся окне Новый запрос из предложенных типов запросов (Конструктор, Простой запрос, Перекрестный запрос, Повторяющиеся записи, Записи без подчиненных) следует выбрать Конструктор.

В окне Добавление таблицы выбрать используемые в запросе таблицы и нажать кнопку [Добавить]. Затем, нажав кнопку [Закрыть], выйти из окна Добавление таблицы.
В результате появится окно конструктора запросов Имя запроса: запрос на выборку.
Окно конструктора запросов разделено на две панели. Верхняя панель содержит схему данных запроса, включающую в себя выбранные для данного запроса таблицы, которые представлены списками полей. Нижняя панель является бланком запроса по образцу (QBE), который нужно заполнить.

Схема данных запроса. В окне запроса отображаются выбранные таблицы и связи между ними, имеющиеся в логической схеме (схеме данных) БД. Кроме того, Access автоматически устанавливает между таблицами дополнительные связи, которых не было в логической модели, в том случае, если таблицы имеют поля с одинаковыми именами и типами данных (атрибутами). Логические связи между таблицами, которые Access не может установить автоматически, может создать пользователь, перетащив задействованные в связи поля из одного списка полей в другой.

При использовании в запросе других запросов или таблиц, не представленных в логической схеме базы данных, с ними также могут быть установлены связи-объединения, т.е. связи без ключевого слова.

Бланк запроса по образцу. Бланк запроса по образцу представлен в нижней панели окна запроса в виде таблицы, которая имеет для заполнения следующие строки: Поле:, Имя таблицы:, Сортировка:, Вывод на экран:, Условие отбора:, или:. До формирования запроса эта таблица не заполнена.

Каждый столбец бланка является одним полем запроса. Эти поля могут использоваться для включения их в таблицу результата выполнения запроса, задания сортировки по ним, а также задания условий отбора записей.

При заполнении бланка запроса необходимо:

- в строку Поле включить имена полей, используемых в запросе;
- в строке Вывод на экран отметить поля, которые должны быть включены в результирующую таблицу;
- в строке Условия отбора задать условия отбора записей;
- в строке Сортировка выбрать порядок сортировки записей результата.

Для включения нужных полей из таблиц БД в соответствующие столбцы запроса можно воспользоваться следующими приемами:

- в первой строке бланка запроса Поле щелчком мыши вызвать появление кнопки списка полей и, воспользовавшись ею, выбрать из списка нужное поле. Список содержит все поля таблиц, представленных в бланке запроса;
- перетащить нужное поле из списка полей таблицы в схеме данных запроса в первую строку бланка запроса.

В списке полей каждой таблицы на первом месте стоит символ звездочка (*), имеющий значение «Все поля таблицы», который выбирается, если в запрос включаются все поля.
Модификация запроса. Для добавления поля в бланк запроса надо перетащить его с помощью мыши из таблицы в схеме данных в нужное место бланка. При этом все столбцы полей справа от него передвинутся на один столбец вправо.

Для удаления поля в бланке запроса надо выделить удаляемый столбец, щелкнув кнопкой мыши, а области маркировки столбца, и нажать клавишу [Del] или выполнить пункт меню Правка\Удалить столбец.

Для перемещения поля в бланке надо выделить соответствующий столбец и перетащить его в новую позицию с помощью мыши. При этом столбец, на место которого перемещен новый столбец, и все столбцы справа от него будут сдвинуты вправо.

Вопрос 2. Формирование логических выражений для создания запроса.

Условия отбора записей могут задаваться для одного или нескольких полей в соответствующей строке бланка запроса. Условием отбора является выражение, которое состоит из операторов сравнения и сравниваемых операторов. В качестве операторов сравнения и логических операторов могут использоваться следующие: =, <, >, < >, Between, In, Like, And, Or, Not, которые определяют операцию над одним или несколькими операндами.

Если условие отбора не содержит оператора, то по умолчанию используется оператор =. В качестве операндов могут использоваться литералы, константы и идентификаторы (ссылки).

Литералом является значение, воспринимаемое буквально, а не как значение переменной или результат вычисления (например, число, строка, дата).

Константами являются не изменяющиеся значения (например, True, Falls, Да, Нет, Null).

Идентификатор представляет собой ссылку на значение поля, элемент управления или свойство. Идентификаторами могут быть, например, имена полей, таблиц, запросов, форм, отчетов, которые должны заключаться в квадратные скобки.

Если необходимо указать ссылку на поле в конкретных таблице, форме, отчете, то перед именем поля ставится имя таблицы, также заключенное в квадратные скобки и отделенное от имени поля восклицательным знаком. Например: [Имя таблицы]! [Имя поля]

Условия отбора, заданные в одной строке, связываются с помощью логической операции И, а заданные в разных строках — с помощью логической операции ИЛИ. Эти операции могут быть заданы явно в условии отбора с помощью операторов AND и ORсоответственно.

Сформировать условие отбора можно с помощью построителя выражения. Перейти в окно ***Построитель выражений*** можно, нажав кнопку [Построитель] на панели инструментов или выбрав команду ***Построить*** в контекстно-зависимом меню. При этом курсор мыши должен быть установлен в ячейке ввода условия отбора.

После ввода выражения в бланк и нажатия клавиши [Enter] Access выполняет синтаксический анализ выражения и отображает его в соответствии с результатами этого анализа.

Для выполнения запроса необходимо на панели инструментов конструктора запросов нажать кнопку [Запуск (!)] или [Представление запроса].

Примеры выражений, используемых в качестве условий отбора, приведены в табл. 2.4.

Примеры выражений, используемых в качестве условий отбора

Поле	**Выражение**	**Описание**
ПунктНазначения	"Москва"	Отображает заказы на доставку товаров в Москву
ПунктНазначения	"Москва" Or "Санкт-Петербургу"	Оператор Or используется для отображения заказов на доставку товаров в Москву или Санкт-Петербург
ДатаОтгрузки	Between #05.01.03# And # 10.01.03 #	Оператор Between ... And используется для отображения заказов на отгрузку товаров не ранее 5 января 2003 г. и не позднее 10 января 2003 г.
ДатаОтгрузки	#2/2/03#	Отображает заказы на отгрузку товаров 2 февраля 2003 г.
СтранаДоставки	In("Россия", "США")	Оператор In используется для отображения заказов на доставку товаров в Россию или США
СтранаДоставки	Not "США"	Оператор Not используется для отображения заказов на доставку товаров во все страны, за исключением США
Имя Клиента	Like "С*"	Отображает заказы на доставку товаров клиентам, имена которых начинаются с буквы С
Название	>=«Н»	Отображает заказы на доставку товаров в фирмы, названия которых начинаются с букв, находящихся в

		диапазоне от Н до Я
ДатаЗаказа	< Date()-30	Функция Date используется для отображения заказов, сделанных более чем за 30 дней
ДатаЗаказа	Year([ДатаЗаказа])=2003	Функция Year используется для отображения заказов, сделанных в 2003 **г.**
ДатаЗаказа	Year([ДатаЗаказа])= Year(Now()) And МогиЬ([ДатаЗаказа])= Month(Now())	Функции Year и Month, а также оператор And используются для отображения заказов текущего года и месяца
ОбластьДос-тавки	Is Null	Отображает заказы для клиентов, у которых поле *ОбластьДоставки* является пустым
ОбластьДост-авки	Is Not Null	Отображает заказы для клиентов, у которых поле *Область Доставки* содержит какое-либо значение
Факс		Отображает заказы для клиентов, у которых нет факсимильного аппарата, т.е. для тех клиентов, у которых поле *Факс* содержит пустую строку, а не значение Null

Сортировка данных. Для удобства просмотра можно сортировать записи в таблице в определенной последовательности. Кнопки сортировки на панели инструментов (или команды меню ***За-писи\Сортировка, Сортировка по возрастанию {Сортировка по убыванию)*** позволяют сортировать столбцы по возрастанию или убыванию. Прежде чем щелкнуть по кнопке сортировки, следует выбрать поля, используемые для сортировки. Современные СУБД (такие, как Access) никогда не сортируют таблицы физически, как это делалось раньше. Средства сортировки данных (а также фильтрации, поиска и замены) реализованы в Access как автоматически создаваемые запросы. Записи таблицы всегда располагаются в файле базы данных в том порядке, в котором они были добавлены в таблицу.

Отбор данных с помощью фильтра. Фильтр — это набор условий, применяемых для отбора подмножества записей. В Accessсуществуют фильтры четырех типов: фильтр по выделенному фрагменту, обычный фильтр, расширенный фильтр и фильтр по вво-**ДУ-**

Фильтрование данных в Access производится с помощью кнопок [Фильтр по выделенному] или [Изменить фильтр] либо команды меню ***Записи\Филыпр, Изменить фильтр.*** После нажатия второй кнопки от таблицы остается одна запись. Каждое поле становится полем со списком (когда в нем находится курсор), в котором можно выбрать из списка все значения для данного поля. После щелчка мышью по кнопке [Изменить фильтр] выбираются записи,

соответствующие измененному фильтру. Еще более сложные условия фильтрации можно задать командой меню ***Записи\Фильтр, Расширенный фильтр.***

Вопрос 3. Элементы языка SQL и запросы в форме SQL

SQL (Structured Query Language) — это язык запросов, который используется при работе с реляционными базами данных в современных СУБД (ORACLE, dBASE IY, dBASE Y, Paradoxe, Access и др.).

Язык SQL стал стандартным языком запросов при работе с реляционными базами данных для архитектуры как файл-сервер, так и клиент-сервер, а также в условиях применения системы управления распределенными БД.

Язык SQL использует ограниченный набор команд, но в то же время — это реляционно полный язык, предназначенный для работы с базами данных, создания запросов выборки данных, выполнения вычислений, обеспечения их целостности. Синтаксис версий языка SQL может в определенной степени различаться для отдельных СУБД. Рассмотрим наиболее общие операторы языка SQL.

Операторы языка SQL для работы с реляционной базой данных

Создание реляционных таблиц. Создание реляционной базы дан-ных означает спецификацию состава полей: указание имени, типа и длины каждого поля (если это необходимо). При этом каждая таблица должна иметь уникальное имя.

Синтаксис оператора создания новой таблицы:

CREATE TABLE таблица (поле1 тип [(размер)] [индекс!.] [, поле2 тип [(размер)] [индекс2] [, ...]] [, составной индекс [, ...]])

Здесь таблица — имя создаваемой таблицы; поле1, поле2 — имена полей таблицы; тип — тип поля; размер — размер текстового поля; индекс1, индекс2 — директивы создания простых индексов; составной индекс — директива создания составного индекса.

Каждый индекс имеет уникальное в пределах данной таблицы имя. Для создания простого индекса используется следующая фраза (размещаемая за именем поля):

CONSTRAINT имя индекса {PRIMARY KEY|UNIQUE| REFERENCES внешняя таблица [(внешнее поле)]}

Директива создания составного индекса (размещаемая в любом месте после определения его элементов) имеет следующий вид:

CONSTRAINT имя {PRIMARY KEY (ключевое1[, ключевое2 [, ...]]) | UNIQUE (уникальное! [, ...]]) | FOREIGN KEY (ссылка1[, ссылка2[, ...]]) REFERENCES внешняя таблица [(внешнее поле1[, внешнее поле2 [, ...]])]}

Значения служебных слов:

UNIQUE — уникальный индекс (в таблице не может быть двух записей, имеющих одно и то же значение полей, входящих в него);

PRIMARY KEY — первичный ключ таблицы, который может состоять из нескольких полей (упорядочивает записи таблицы);

FOREIGN KEY — внешний ключ для связи с другими таблицами (может состоять из нескольких полей);

REFERENCES — ссылка на внешнюю таблицу.

Пример 1 Создание таблицы:

CREATE TABLE Студент

([Имя] TEXT,

[Фамилия] TEXT,

[Дата рождения] DATETIME,

CONSTRAINT Адр UNIQUE ([Имя]), [Фамилия], [Дата рождения]))

В результате выполнения данного запроса будет создана таблица СТУДЕНТ, имеющая в своем составе:

два текстовых поля — Имя, Фамилия;

одно поле типа дата/время — Дата рождения.

Будет также создан составной индекс с именем Адр по значениям указанных полей, который будет иметь уникальное значение, так как в таблице не может быть двух записей с одинаковыми значениями полей, образующих его.

Изменение структуры таблиц. При необходимости можно изменить структуру таблицы:

удалить существующие поля;

добавить новые поля;

создать или удалить индексы.

При этом все указанные действия затрагивают только одно поле или один индекс:

ALTER TABLE таблица

ADD{[COLUMN]поле тип[(размер)][CONSTRAINT индекс] CONSTRAINT составной индекс}|

DROP {[COLUMN] поле i CONSTRAINT имя индекса}}

Опция ADD обеспечивает добавление поля таблицы, а опция DROP — удаление. Добавление опции CONSTRAINT означает подобные действия для индексов таблицы.

Пример 2 Изменение структуры таблицы:

ALTER TABLE Студент ADD COLUMN [Группа] TEXT(5)

Для создания нового индекса к существующей таблице можно также использовать следующую команду:

CREATE [UNIQUE] INDEX индекс ON таблица (поле[, ...])

[WITH {PRIMARY I DISALLOW NULL|IGNORE NULL}]

Фраза WITH обеспечивает наложение условий на значения полей, включенных в индекс:

DISALLOW NULL — запретить пустые значения в индексированных полях новых записей;

IGNORE NULL — включать в индекс записи, имеющие пустые значения в индексированных полях.

Пример 3 Создание индекса таблицы:

CREATE INDEX Гр ON Студент([группа]) WITH DISALLOW NULL

Удаление таблицы. Для удаления таблицы (одновременно и структуры, и данных) используется следующая команда:

DROP TABLE имя таблицы

Для удаления только индекса таблицы (сами данные при этом не разрушаются) необходимо выполнить следующую команду:

DROP INDEX имя индекса ON имя таблицы

Пример 4 Удаление только индекса Адр таблицы: DROP INDEX Адр ON Студент Удаление всей таблицы:

DROP TABLE Студент

Ввод данных в таблицу. Формирование новой записи в таблице выполняется следующей командой:

INSERT INTO таблица [(поле1[, поле2 [, ...]])] VALUES (значение!. [, значение2[, ...])

Здесь указываются имя таблицы, в которую добавляют запись, и состав полей, для которых вводятся значения.

Пример 5 Ввод данных в таблицу:

INSERT INTO Студент ([Фамилия], [Имя], [Дата рождения]) VALUES ("Петров", "Иван", 23/3/80)

Возможен также групповой ввод записей (пакетный режим), являющихся результатом выборки (запроса) из других таблиц:

INSERT INTO таблица [IN внешняя база данных] SELECT [источник.] поле1 [, поле2[, ...] FROM выражение WHERE условие

В этом случае сначала выполняется оператор подзапроса SELECT, который и формирует выборку для добавления данных.

Фраза SELECT определяет структуру данных источника пере-даваемых записей — имена таблицы и полей, содержащих исходные данные для загрузки в таблицу.

Оператор FROM позволяет указать имена исходных таблиц, участвующих в формировании выборки, а фраза WHERE задает условия выполнения подзапроса. При этом структура данных выборки должна соответствовать структуре данных таблицы, в которую производится добавление.

Добавление (перезагрузка) записей возможно и во внешнюю базу данных, для которой указывается полностью специфицированное имя (диск, каталог, имя, расширение). При этом структуры таблиц должны совпадать.

Пример 6 Ввод данных в таблицу:

INSERT INTO Студент SELECT [Студент-заочник].* FROM [Студент-заочник]

Этой фразой все записи таблицы СТУДЕНТ-ЗАОЧНИК будут добавлены в таблицу СТУДЕНТ.

Пример 7 Ввод данных в таблицу:

INSERT INTO Студент SELECT [Студент-заочник].* FROM [Студент-заочник] WHERE [Дата рождения] > = # 01/01/80 #

В этом случае записи таблицы СТУДЕНТ-ЗАОЧНИК добавляются в таблицу СТУДЕНТ, если дата рождения студента больше или равна ука-занной.

Операции соединения таблиц. Операцию INNER JOIN можно использовать в любом предложении FROM. Она создает симметричное объединение — наиболее частую разновидность внутреннего объединения.

Записи из двух таблиц объединяются, если связующие их поля содержат одинаковые значения:

FROM таблица1 INNER JOIN таблица2 ON таблица!., поле1=таблица2.поле2

Данный оператор описывает симметричное соединение двух таблиц по ключам связи (поле1; поле2). Новая запись формируется в том случае, если в таблицах содержатся одинаковые значения ключей связи.

Возможны следующие варианты операции соединения таблиц: LEFT JOIN (левостороннее) соединение, когда выбираются все записи левой таблицы и только те записи правой таблицы, которые содержат соответствующие ключи связи;

RIGHT JOIN (правостороннее) соединение, когда выбираются все записи правой таблицы и только те записи левой таблицы, которые содержат соответствующие ключи связи.

Пример 8 Соединение таблиц:

SELECT Студент.*, Оценка.* FROM Студенты INNER JOIN Оценка ON Студент.[№ зач.книжки] = Оценка.[№ зач. книжки]

SELECT Студент.*, Оценка.* FROM Студенты LEFT JOIN Оценка ON Студент. [№ зач.книжки] = Оценка. [№ зач. книжки]

SELECT Студент.*, Оценка.* FROM Студенты RIGHT JOIN Оценка ON Студент. [№ зач.книжки] = Оценка. [№ зач. книжки]

В первом случае создается симметричное соединение двух таблиц по полю [№ зач.книжки]. При этом записи не выводятся, если значение их ключей связи (указанное поле) не представлено в двух таблицах.

Во втором случае выводятся все записи таблицы СТУДЕНТ и соответствующие им записи таблицы ОЦЕНКА.

В третьем случае, наоборот, выводятся все записи таблицы ОЦЕНКА и соответствующие им записи таблицы СТУДЕНТ.

Операции JOIN могут быть вложенными для последовательного соединения нескольких таблиц.

Пример 9 Соединение таблиц:

SELECT Студент.*, Оценка.*, Дисциплина.[Наименование дисциплины] FROM (Студент INNER JOIN (Оценка INNER JOIN (Дисциплина ON Оценка. [Код дисциплины] = Дисциплина.[Код дисциплины]) ON Студент.[№ зач.книжки] = Оценка.[№ зач.книжки])

Здесь сначала происходит соединение таблиц ОЦЕНКА и ДИСЦИП-ЛИНА по ключу связи [Код дисциплины]. Соединение симметричное, т.е. если коды дисциплины не совпадают, записи этих таблиц не соединяются. Затем происходит соединение таблиц СТУДЕНТ и ОЦЕНКА по ключу связи [№ зач.книжки].

Таким образом, при условии совпадения ключей связи на выходе запроса получается результат соединения трех таблиц.

Удаление записей в таблице. В исходной таблице можно удалить отдельные записи или все записи, сохранив при этом ее структуру и индексы. При удалении записей в индексированной таблице автоматически корректируются ее индексы:

DELETE [таблица.*] FROM выражение WHERE условия отбора

Полная чистка таблицы от записей и очистка индексов выполняются следующей операцией:

DELETE * FROM таблица

Пример 10 Удаление всех записей в таблице:

DELETE * FROM Студент

Удаление только тех записей, в которых поле [Дата рождения] больше указанной даты:

DELETE * FROM Студент WHERE [Дата рождения]> #1.1.81 #

Удаление в таблице записей, связанных с другой таблицей (условия удаления записей могут относиться к полям связанных таблиц):

DELETE таблица.* FROM таблица INNERJOIN другая таблица ON таблица.[поле N] = [другая таблица].[поле M] WHERE условие

Пример 11 Удаление записей в таблице СТУДЕНТ, для которых имеются связанные записи в таблице СТУДЕНТ-ЗАОЧНИК:

DELETE Студент.* FROM Студент INNER JOIN [Студент-заочник] ON Студент.[Группа] = [Студент-заочник]. [Группа]

Обновление (замена) значений полей записи. Изменение значений нескольких полей одной записи или группы записей таблицы, удовлетворяющих условиям отбора производится следующей фразой:

UPDATE таблица SET новое значение WHERE условия отбора Новое значение указывается как имя поля=новое значение

Пример 12 Отбор студентов, чьи фамилии начинаются на букву В и дата рождения не превышает указанной, для перевода в группу 1212:

UPDATE Студент SET [Группа] = "1212"

WHERE [Фамилия] LIKE 'В*' AND [Дата рождения] < = #01/01/81*

Изменение в таблице СТУДЕНТ номеров групп обучения путем добавления к ним буквы «а», если эти группы встречаются в таблице СТУ-ДЕНТ-ЗАОЧНИК:

UPDATE Студент INNER JOIN [Студент-заочник] ON Студент. [Группа] = [Студент-заочник].[Группа] SET [Группа] = [Группа]

Контрольные вопросы:

1. Формирование логических выражений для выбора информации.
2. Задание вариантов связей между парами таблиц данных, определяющих результаты выдачи.
3. Упорядочение результатов запросов.
4. Ограничение количества выдаваемых записей.
5. Ввод параметров запросов и получения результатов в виде таблиц данных.

Литература

1. Попов В.Б. Основы информационных и телекоммуникационных технологий.- Финансы и статистика, 2007 г. – 336 с.
2. Угринович Н.Д. Информационные технологии и компьютеризация делопроизводства. – М.:БИНОМ ЛЗ.,2007.-394с.
3. Шамраев А.В. Правовое регулирование информационных технологий (анализ проблем и основные документы). – Статут, 2003. – 1013 с.
4. Хроленко А.Т., Денисов А.В. Современные информационные технологии для гуманитария. – Флинта, 2007. – 128 с.
5. Федотова Е.Л. Информационные технологии в профессиональной деятельности. – Форум, 2008. – 368 с.
6. Емельянова Н.З., Партыка Т.Л., Попов И.И. Основы построения автоматизированных информационных систем. – Форум, 2007. – 416 с.
7. Беляева Т. М., Важнов С. А., Вешняков В. В., Кудинов А. Т., Пальянова Н. В. Основы информатики и математики для юристов. – Элит, 2007. – 368 с.
8. Казанцев С.Я. Информатика и математика для юристов. Учебник. – Юнити, 2008. – 560 с.
9. Строганов М.П., Щербаков М.А. Информационные сети и телекоммуникации. – Высшая школа, 2008. – 151 с.
10. Микрюков В.Ю. Информация, информатика, компьютер, информационные системы, сети. – Феникс, 2007. – 448 с.