

УНИВЕРСИТЕТ ГРАЖДАНСКОЙ ЗАЩИТЫ УКРАИНЫ

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И СИСТЕМ УПРАВЛЕНИЯ**

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**К ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ И ЛАБОРАТОРНЫХ РАБОТ
ПО КУРСУ**

**«АСУ и информационное обеспечение
деятельности службы гражданской защиты»**

**Раздел: «Проектирование и создание информационных
систем обработки данных в АСУ МЧС»**

*для магистров специальности «Пожарная безопасность»
всех форм обучения*

Харьков
2007

Утверждено на заседании кафедры информационных технологий и систем управления
Протокол № 12 от 03.07.2007

Методические рекомендации к выполнению контрольной и лабораторных работ по курсу «АСУ и информационное обеспечение деятельности службы гражданской защиты»: Раздел: «Проектирование и создание информационных систем обработки данных в АСУ МЧС» для магистров специальности «Пожарная безопасность» всех форм обучения. / Сост. Шматко А.В., Щербак Г.В. – Харьков: УЦЗУ, 2007.

ВВЕДЕНИЕ

Базы данных стали основой информационных систем и в корне изменили методы работы службы гражданской защиты. В частности, развитие данной технологии в последнее время привело к созданию весьма мощных и интуитивно понятных систем сбора и анализа данных для автоматизированных систем управления (АСУ) в подразделениях МЧС Украины. Все это сделало системы баз данных (БД) доступными широкому пользователю. К сожалению, кажущаяся простота данных систем способствовала тому, что пользователи стали самостоятельно создавать базы данных, не имея достаточно знаний о методах проектирования эффективно работающих систем. Целью данных методических рекомендаций является ясное и четкое изложение практических основ теории баз данных, в частности, конкретная методология проектирования БД, предназначенная для использования, как профессиональными разработчиками, так и не профессионалами (пользователями).

Предложенная в этих методических рекомендациях методология работы с реляционными системами управления базами данных (СУБД), доминирующими в настоящее время в практических приложениях, прошла проверку в академической среде. Проектирование баз данных состоит из трех фаз: концептуальной, логической и физической. Первая фаза предусматривает создание концептуальной модели данных, не зависящей от каких-либо физических характеристик. Во второй фазе, назначение которой состоит в создании логической модели данных, концептуальная модель подвергается доработке посредством удаления элементов, которые не могут быть реализованы в реляционных системах. В третьей фазе логическая модель данных преобразуется в физический проект, предназначенный для реализации в среде конкретной СУБД.

Каждая фаза предлагаемой методологии представлена в виде последовательности этапов. Для облегчения процесса изучения методологии и понимания некоторых важных вопросов, в трех разделах этих рекомендаций дается подробное описание практического примера — учебного проекта, получившего название Информационная система «Нормативные документы». Помимо этого в приложении 1 приводится ряд примеров учебных информационных систем для самостоятельного проектирования.

РАЗДЕЛ 1. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ И БАЗ ДАННЫХ

1.1 Этапы проектирования базы данных

При проектировании базы данных решаются три основных проблемы:

- Как адекватно отразить предметную область и информационные потребности пользователей в концептуальной модели? Эту проблему называют проблемой инфологического проектирования баз данных. Цель инфологического этапа проектирования состоит в получении семантических (смысловых) моделей, отражающих информационное содержание проблемы.
- Каким образом отобразить объекты предметной области в абстрактные объекты модели данных так, чтобы это отображение не противоречило семантике предметной области, и было, по возможности, наилучшим (эффективным, удобным и т.д.)? Эта проблема известна как проблема логического проектирования баз данных. Цель логического этапа проектирования - организация данных, выделенных на предыдущем этапе, в форму, принятую в выбранной СУБД.
- Как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создание каких дополнительных структур (например, индексов) необходимо потребовать и т.д.? Эту проблему называют проблемой физического проектирования баз данных. Цель физического этапа - выбор рациональной структуры хранения данных и методов доступа к ним.

На рис.1 схематически представлены этапы проектирования БД.

1.2. Инфологическое проектирование

Одной из наиболее популярных семантических моделей данных на этапе инфологического проектирования является неформальная модель "**Сущность-Связь**" (Entity-Relationship - ER-модель). Модель была предложена Ченом (Chen) в 1976 г. Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов.

1.2.1. Сущности и атрибуты

Сущность (объект) - это реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступна. Различают такие понятия как **тип сущности** и **экземпляр сущности**. Понятие **тип сущности** относится к набору однородных предметов, событий, личностей, выступающих как единое целое. **Экземпляр сущности** относится к конкретной вещи в наборе. Для идентификации конкретных экземпляров сущности используются **атрибуты**, выполняющие роль идентификаторов.

Атрибут - поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества значений. Каждый атрибут обеспечивается именем, уникальным в пределах сущности

(<имя сущности>.<Имя атрибута>).

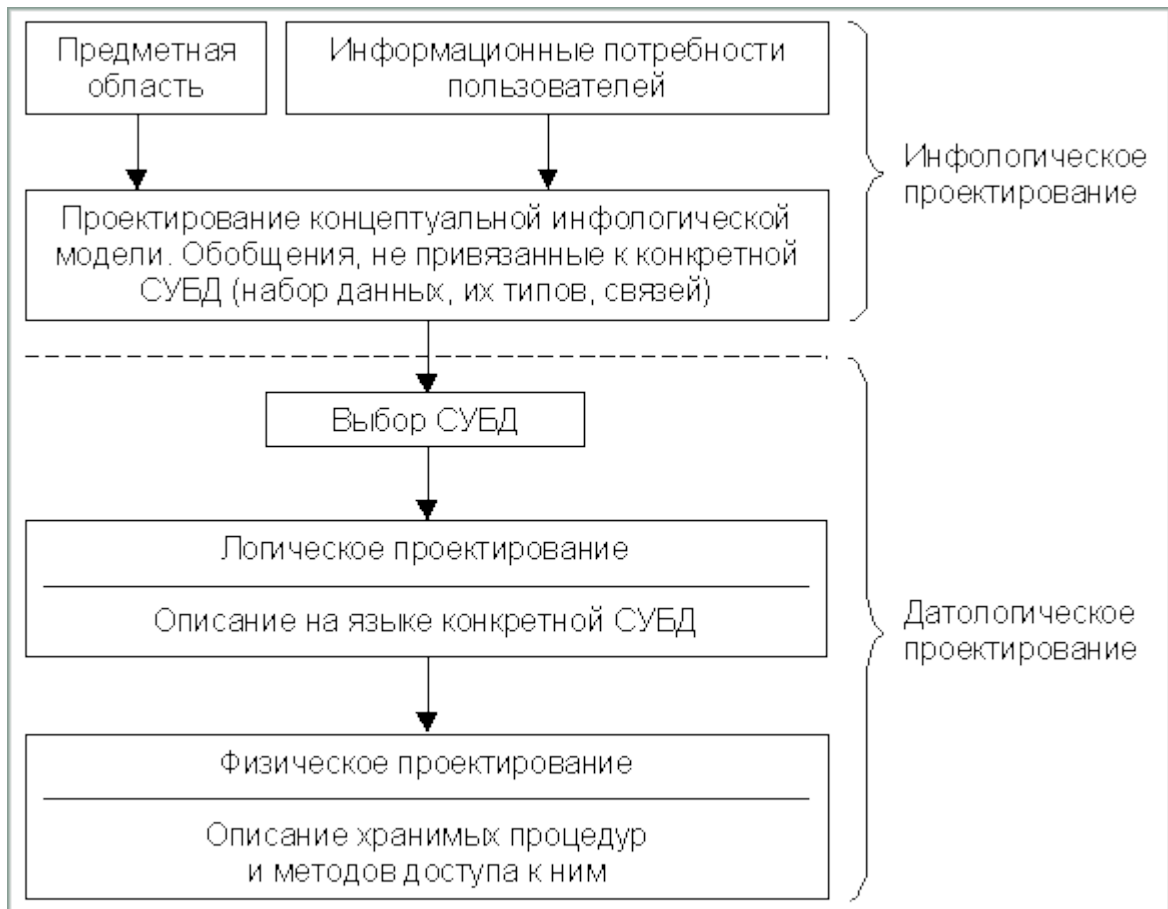


Рис.1. Этапы проектирования БД

Например,
Самолет.Размах крыла
Кошка.Вес
Документ.Тип Документа

Диапазон допустимых значений, которые может принимать атрибут, называется **доменом**

Замечание. Абсолютное различие между **типами сущностей** и **атрибутами** отсутствует. **Атрибут** является таковым только в связи с типом сущности.

Множество из одного или нескольких атрибутов, значения которых однозначно определяют каждый экземпляр сущности, называются **идентификатором (ключом)**. Каждая сущность должна иметь хотя бы один идентификатор (рис.2). Если идентификаторов несколько, один из них выбирается как **привилегированный (первичный)**.

Атрибуты могут классифицироваться по принадлежности к одному из трех различных типов:

- описательные;
- указывающие;
- вспомогательные.

На рис.3 представлены возможные варианты представления сущностей. Мы будем пользоваться графическим методом.

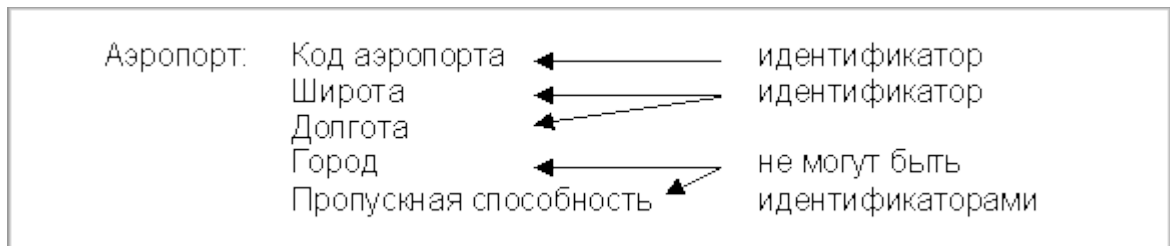


Рис.2. Пример идентификаторов

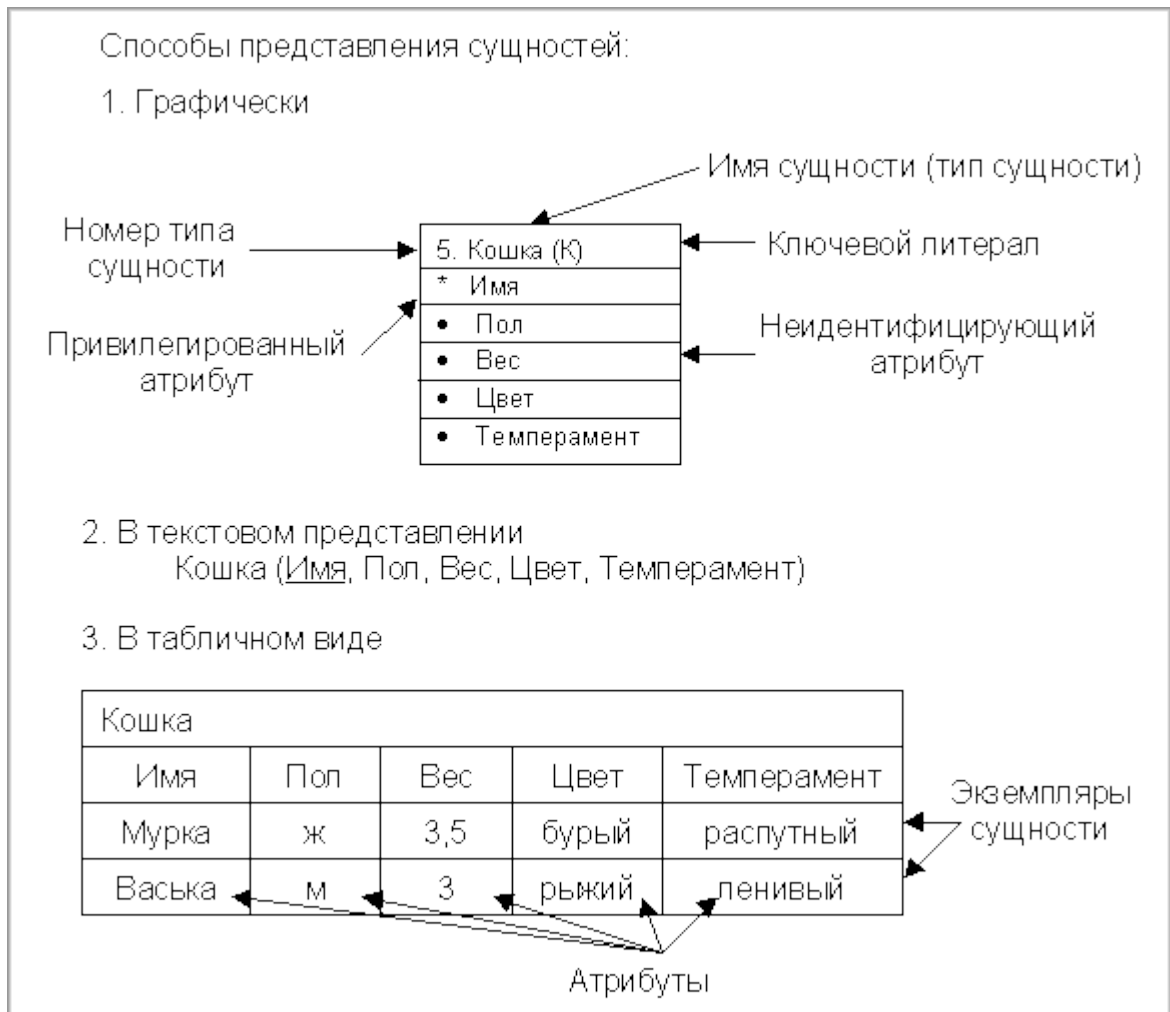


Рис.3. Способы представления сущностей

Описательные атрибуты представляют характеристики, внутренне присущие каждому экземпляру сущности:

Счет.Сальдо

Источник электроснабжения.Полярность

Кошка.Вес

Если значение описательного атрибута изменяется, то это говорит о том, что некоторый аспект экземпляра сущности изменился, но сам экземпляр остался прежним (вес Кошки изменился, сама Кошка осталась прежней).

Указывающие атрибуты используются для присвоения имени или обозначения экземпляров сущности:

Счет.Номер

Груз.Номер накладной

Город.Название

Изменение значения указывающего атрибута говорит о том, что данному экземпляру дается новое имя. Указывающие атрибуты часто используются как идентификатор или как часть идентификатора.

Вспомогательные атрибуты используются для связи экземпляра одной сущности с экземпляром другой:

Кошка.Имя хозяина

Счет.ID клиента

Магнит.Источник электроснабжения

Если значение вспомогательного атрибута меняется, это означает, что теперь другие экземпляры связаны между собой (изменение источника электроснабжения PS10 на PS12 определяет, что магнит связан с другим источником).

Правила атрибутов:

1. Один экземпляр сущности имеет одно единственное значение для каждого атрибута в любое данное время. В табличной интерпретации это правило требует, чтобы существовал один и только один элемент данных для каждой строки и столбца. Это делает невозможным появление незаполненных ячеек и ячеек с группой значений (рис.4).

Владелец	Модель	Изготовитель
Грин	Фургон	Шевроле
Джонс	Джип	
Кларк	Грузовик	Форд Сузуки

Нарушение первого правила атрибутов (незаполненные ячейки и ячейки с группой значений)

Рис.4. Первое правило атрибутов

2. Атрибут не должен содержать никакой внутренней структуры (рис.5)

Заводской номер	Модель изготовителя
670978	Москвич 2141
879800	Москвич 412
078876	Лѳаз 969M

Нарушение второго правила атрибутов (невозможно выделить изготовителя)

Рис.5. Второе правило атрибутов

3. Когда сущность имеет составной идентификатор, каждый атрибут, не являющийся частью идентификатора, представляет характеристику всей сущности, а не ее части, а тем более не характеристику чего-либо другого (рис.6.).

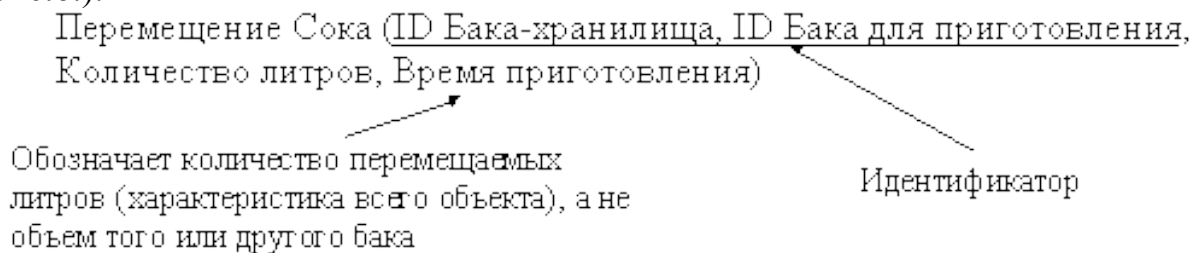


Рис.6. Третье правило атрибутов

4. Каждый атрибут, не являющийся частью идентификатора, представляет характеристику экземпляра, указанного идентификатором, а не характеристику некоторого другого атрибута – неидентификатора (рис.7.).

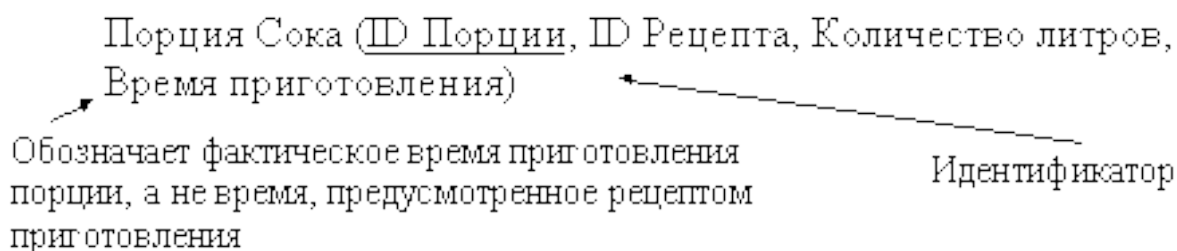


Рис.7. Четвертое правило атрибутов

1.2.2. Связи

Связь (Relationship) - это поименованная графически изображаемая ассоциация, устанавливаемая между сущностями и представляющая собой абстракцию набора отношений, которые систематически возникают между различными видами предметов в реальном мире. Существуют три фундаментальных вида связи: **один к одному (1:1)**, **один ко многим (1:M)**, **многие ко многим (M:N)**. Эти фундаментальные виды связей относятся к числу безусловных связей и требуют участия каждого экземпляра сущности.

Связь один к одному (1:1) существует, когда один экземпляр одной сущности связан с единственным экземпляром другой сущности (рис.9).

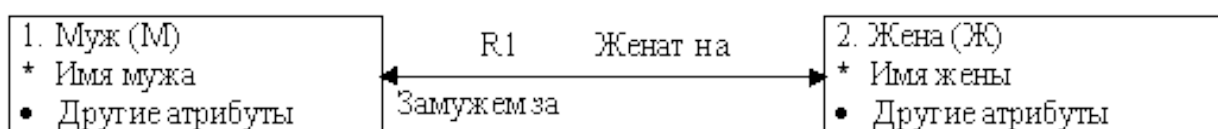


Рис.9. Связь один к одному (1:1)

Связь один ко многим (1:M) существует, когда один экземпляр одной сущности связан с одним или более экземплярами другой сущности и

каждый экземпляр второй сущности связан только с одним экземпляром первой сущности (рис.10).

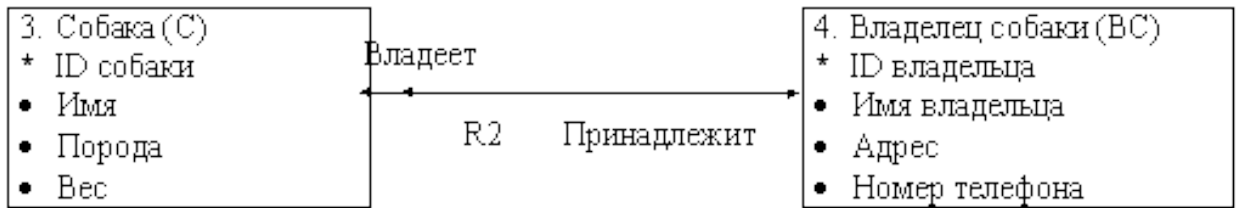


Рис.10. Связь один ко многим

Связь многие ко многим (M:N) существует, когда один экземпляр одной сущности связан с одним или более экземплярами другой сущности и каждый экземпляр второй сущности связан с одним или более экземплярами первой сущности (рис.11).

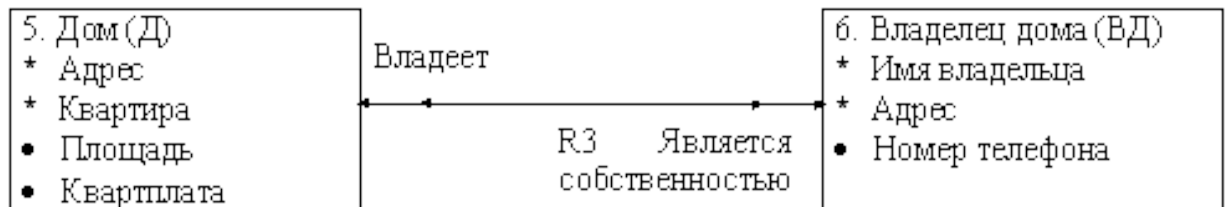


Рис.11. Связь многие ко многим

1.3. Проектирование реляционных баз данных.

Для того чтобы спроектировать реляционную БД нужно выделить определенную совокупность таблиц, которые содержат нужную информацию, и установить связи между этими таблицами. Для того, чтобы спроектировать БД, таким образом, применяют два подхода: «сверху вниз» или «снизу вверх». При первом подходе сначала определяются основные задачи, для решения которых строится БД и потребности этих задач в определенных данных. А уже потом эти данные распределяются по таблицам и связываются между собой. При втором подходе изучается предметная область, реквизиты всех документов, проводится анализ данных и устанавливаются типичные объекты этой области. После чего строятся реляционные таблицы и связи между ними. В целом, процесс проектирования БД распределяется на такие этапы:

1 этап. Формирование задач, решение которых необходимо при работе БД — по ведению информации, выборкам, созданию отчетов и т.п. На этом этапе, прежде всего, учитываются уже существующие документы (накладные, расчеты, бланки и т.д.)

2 этап. Анализ данных. Определяются данные, которые должны находиться в БД и обеспечивать выполнение необходимых задач. Эти

данные, как правило, представлены в виде реквизитов, которые содержатся в разнообразных документах–источниках БД.

3 этап. Определение структуры данных. На этом этапе все данные распределяются по объектам, и эти объекты сопоставляются с таблицами БД. Для каждого из данных надо указать его имя в таблице, тип и формат. В реляционной БД на этом этапе проводят **процесс нормализации**, с помощью которого выбирается наиболее эффективный и гибкий способ сохранения информации.

Правила нормализации:

- Каждое поле любой таблицы должно быть уникальным (не дублировать данные).
- Информационный объект должен иметь уникальный идентификатор – **первичный ключ** (простой или составной). Например, для списка студентов фамилия студента не может быть первичным ключом, так как может дублироваться, поэтому прибавляют идентификатор – номер зачетной книжки.
- Все не ключевые поля должны быть независимы.
- Все поля, которые входят в составной ключ, тоже должны быть взаимно независимыми.
- Каждому значению первичного ключа должно отвечать только одно значение не ключевого поля, и это значение должно относиться к объекту таблицы.

4 этап. Формирование связей между таблицами БД.

РАЗДЕЛ 2. ПРИМЕР РАЗРАБОТКИ ЛОГИЧЕСКОГО ПРОЕКТА БАЗЫ ДАННЫХ

Остановимся на методологии прохождения этапов на примере автоматизации ведения нормативных документов.

1 этап. На первом этапе формируем перечень задач.

В нашем примере нужно построить базу данных, работая с которой пользователь сможет решать такие задачи:

- получать справки о любом документе: дата издания документа, тип документа, текст документа.
- для каждого из типов документов необходимо сохранять и при потребности выдавать список документов, дату их опубликования, текст документа.
- осуществлять выборки документов: по типу, по дате, по наименованию.

2 этап. Для каждой из задач, которые были выделены на первом этапе, заполняем бланк следующего вида

1).

Наименование задачи	Получение сведений о документах
Краткое описание:	Ввод, сохранение и выдача информации о документах
Описание атрибута	Объект
Название документа	Документ
Тип документа	Тип
Дата издания	Реестр
Текст документа	Текст

2).

Наименование задачи	Получение справок о типе документов
Краткое описание:	Ввод, сохранение и выдача информации о типах документов
Описание атрибута	Объект
Название документа	Документ
Тип документа	Тип
Текст документа	Текст
Дата издания	Реестр

3).

Наименование задачи	Выборки документов
Краткое описание:	Выборки документов по определенным критериям
Описание атрибута	Объект
Название документа	Документ
Тип документа	Тип
Текст документа	Текст
Дата издания	Реестр

3 этап. На этом этапе осуществляется переход от объектов к таблицам. С этой целью для каждого объекта, которые были выделены на этапе 2, надо заполнить определенный бланк, на базе которого будут построены таблицы. В нашем случае мы выделили такие объекты: **ДОКУМЕНТ, ТИП, РЕЕСТР**. Перед созданием таблиц необходимо провести анализ данных и выполнить нормализацию данных. Требования нормализации требуют, например, для объекта **ДОКУМЕНТ** ввести дополнительное поле **Код документа**.

С учетом этих замечаний бланк объекта **ДОКУМЕНТ** будет иметь такой вид:

Имя объекта	ДОКУМЕНТ		
Краткое описание	Данные о документе		
Описание атрибута	Имя атрибута	Тип и формат	Тип ключа
Код документа	Код документа	Текстовый (50)	Первичный
Наименование документа	Наименование док	Текстовый (255)	
Текст документа	Текст док	Поле МЕМО	

Далее строится таблица связей.

Имя атрибута	Имя связанной таблицы	Имя связанного атрибута	Тип связи
Код документа	РЕЕСТР	Код документа	1:M

В этой таблице (связей данных объекта) в первой колонке расположены имена тех полей текущей таблицы **ДОКУМЕНТ**, которые являются внешними ключами. Вторая, третья и четвертая колонки этой таблицы могут быть заполнены позднее, после описания всех других объектов базы данных. При этом во второй колонке располагается имя таблицы **РЕЕСТР**, с полем которой связано поле внешнего ключа текущей таблицы **ДОКУМЕНТ**.

Примечания.

1. На этом этапе таблицам и данным предоставляются определенные имена с учетом требований к именам в конкретной БД.
2. На этом этапе для каждого атрибута выбираются его тип и размер.
3. Ячейку **Тип ключа** заполняем для тех полей, которые входят в состав какого-нибудь ключа.

Аналогично строится бланк других объектов.

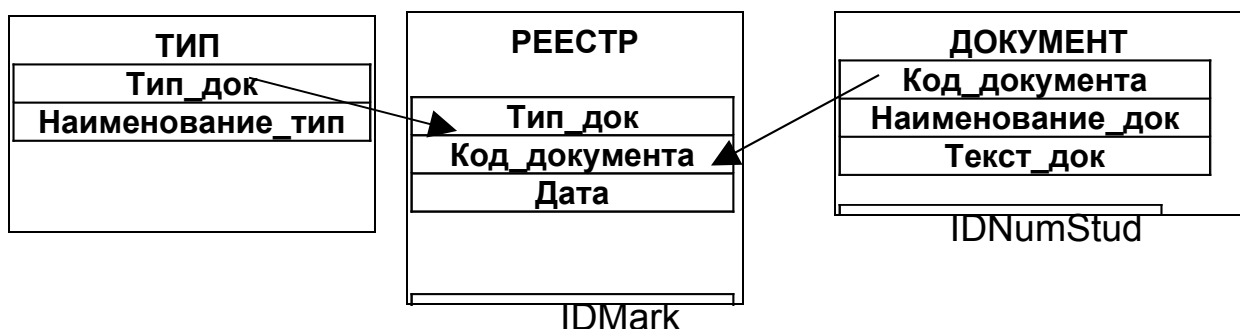
Имя объекта	ТИП		
Краткое описание	Данные о типе документов		
Описание атрибута	Имя атрибута	Тип и формат	Тип ключа
Тип документа	Тип_док	Текстовый (50)	Первичный
Наименование типа	Наименование_тип	Текстовый (255)	

Имя атрибута	Имя связанной таблицы	Имя связанного атрибута	Тип связи
Тип_док	РЕЕСТР	Тип_док	1:M

Имя объекта	РЕЕСТР		
Краткое описание	Данные о реестре документов		
Описание атрибута	Имя атрибута	Тип и формат	Тип ключа
Тип документа	Тип_док	Текстовый (50)	Внешний
Код документа	Код_документа	Текстовый (50)	Внешний
Дата издания	Дата	Дата/время	

Имя атрибута	Имя связанной таблицы	Имя связанного атрибута	Тип связи
Тип_док	ТИП	Тип_док	M:1
Код_документа	ДОКУМЕНТ	Код_документа	M:1

4 этап. На основе информации о связях объектов, которая была сформирована на третьем этапе, строим схему связей.



РАЗДЕЛ 3. ПРИМЕР РАЗРАБОТКИ ФИЗИЧЕСКОГО ПРОЕКТА БАЗЫ ДАННЫХ СРЕДСТВАМИ СУБД ACCESS

3.1. Основные возможности СУБД Access

СУБД реляционного типа Microsoft Access появилась в составе системы MS Office в 1992 г. Как и другие СУБД, она позволяет создать БД и совершать с ней различные операции: обрабатывать данные, составлять всевозможные отчеты. При этом нет необходимости в дополнительном написании программы-интерфейса.

Основные области применения:

- 1) Малый бизнес (бухгалтерский учет, конторский труд, торговля и др.);
- 2) Рабочие группы внутри больших корпораций (обмен данными внутри группы и между ними);
- 3) Создание персональных БД (справочники телефонов, адресов, каталоги книг, учет личных расходов, наборы рецептов и т.д.).

Основным объектом БД Access является **таблица**. Причем, система имеет средства, позволяющие обрабатывать несколько связанных таблиц в одной БД. Это дает возможность распределять данные по таблицам в соответствии с их содержанием. Редактирование данных в одном месте БД приводит к изменению данных в другом.

Управление данными осуществляется с помощью меню, панелей инструментов, контекстных меню, указателей мыши. Кроме того, Access имеет большое количество программ, которые автоматизируют процесс создания и ведения БД. Они называются "**мастерами**". Имеются удобные средства для копирования и перемещения строк, полей и фрагментов из одной таблицы в другую, получения копий таблиц и их структур. Можно также копировать и перемещать данные из любого приложения MS Windows в MS Access и наоборот.

3.1.1 Состав БД Access

Для решения задачи в определенной предметной области СУБД Access создает файл базы данных с расширением *.mdb.

В БД, помимо таблиц, в которых содержатся сами данные, могут содержаться также и средства работы с ними — запросы, формы, отчеты, макросы и модули. Все эти элементы называются объектами Access.

Таблица — объект, который служит для определения и хранения данных одного вида (например, цены товаров, сведения о клиентах и т.п.).

Каждая строка таблицы называется **записью**, а каждый столбец — **полем**. Каждая запись содержит данные об одном объекте.

Поле (совокупность полей), которое однозначно определяет запись, называется **ключевым**. **Ключевым** обычно назначается поле, все значения которого в БД являются уникальными. Например, код документа, табельный номер и т.п. Фамилию не всегда можно использовать в качестве ключевого поля, т.к. в одной БД могут быть однофамильцы.

Поле (совокупность полей), по которому осуществляется упорядочение записей (по возрастанию или убыванию) называется **индексом**. Значения индексных полей могут совпадать. Поэтому любой ключ может быть и индексом, но не наоборот.

Использование ключей и индексов ускоряет поиск данных в БД. Их также используют при связывании таблиц.

Запросы позволяют получать на экране разные сведения из таблиц. Пользователь формирует условия запроса, в соответствии с которыми из базы выбираются необходимые данные. При этом сведения выдаются в виде таблицы.

Если требуется представить сведения (данные) в другом формате, то используют выдачу данных в виде **форм**. Access содержит три стандартного вида формы: **в столбец**, **ленточную** и **табличную**. Кроме того, можно создать форму с помощью **мастера форм**. Формы позволяют вводить и редактировать данные, выполнять вычисления, использовать фильтры для отбора информации.

Данные можно представить и в виде **отчета**. В этом случае данные представляются в любом формате с различной степенью детализации. Кроме того, в **отчетах** можно отобразить и печатать записи в группах. В **отчет** можно добавлять линии, рамки, рисунки. Форматы отчетов соответствуют стандартным форматам бумажного листа. Таким образом, они лучше подходят для визуального и печатного представления данных. Однако в отчете нельзя выполнить редактирование данных.

Макрос – объект, предназначенный для автоматизации нескольких действий, которые ранее выполнялись с помощью команд меню или кнопок панели инструментов Access.

Модуль – объект, представляющий собой программу, написанную на языке Visual Basic. Используется для создания приложений.

Каждый объект создается **Конструктором** или **Мастером**. Создание объекта в первом случае состоит в ручном построении с нулевого уровня. Во втором — состоит из ряда шагов. После ответов на поставленные вопросы **Мастер** строит прототип объекта. В дальнейшем его можно улучшить с помощью **Конструктора**.

3.1.2. Типы данных

Тип данных представляет собой множество значений, которые может принимать поле записи. В поле могут содержаться значения из множества того типа, которому оно принадлежит.

В Access используются следующие типы данных.

1.Текстовый тип. Используется для хранения любой последовательности символов в пределах от 1 до 255. Это может быть последовательность, состоящая из букв, цифр, знаков пунктуации, пробелов, специальных символов и т.п.

2. Числовой тип. Используется для хранения числовых значений (кроме денежных сумм). Обычно содержит значения, над которыми впоследствии выполняют вычисления. Числа могут содержать десятичные точки.

3. Дата/время. Используется для хранения календарных дат в пределах 100-9999 год и (или) времени суток в 12-ти или 24-х часовом формате. Значение даты и времени можно хранить и в текстовом поле. Преимущество данного типа: автоматический контроль правильности ввода, а также возможность выполнять вычисления и сортировку.

4. Денежный тип. Используется для хранения значений денежных сумм. Содержит до 4-х знаков в дробной части. Можно использовать в финансовых вычислениях.

5. Счетчик. Используется для автоматической нумерации добавляемой записи, начиная с единицы. Это номер записи таблицы. Изменить его значение нельзя. При удалении записи повторное использование номера невозможно.

6. Логический тип. Используют для хранения логических величин, которые принимают значения «ДА» или «НЕТ» („Истина» или „Ложь»).

7. Поле МЕМО. Используется для хранения текста большого объема. В нём может содержаться до 32 тыс. символов. Это, например, тексты документов, приказов, ГОСТов и т.п.

8. Поле объекта OLE. Содержит объект OLE (Objects Linking and Embedding — объекты связанные и внедренные), т.е. документ другого приложения Windows. Он может содержать аудио- и видеозаписи, таблицы или текстовые документы. Формируется по команде **Вставка – Объект**.

9. Поле типа Гиперссылка. Содержит буквенно-цифровой идентификатор — адрес гиперссылки, который указывает путь к другому объекту или документу. Например, к файлу в локальной сети.

10. Мастер подстановок. Используется для создания связи поля с набором значений, которые может принимать данное поле. При вводе значений такое поле имеет раскрывающийся список для выбора нужного значения. Например, имеется таблица, которая должна содержать 2-х буквенные коды. При ее создании можно указать, что некоторое поле должно содержать только двухбуквенные коды, перечисленные во вспомогательной таблице.

3.2. Создание базы данных

Цели работы:


- познакомиться с основными понятиями БД;
- научиться создавать БД в режиме **Конструктор**;
- научиться вводить и редактировать записи БД в режиме **Таблица**;
- познакомиться с сортировкой БД;
- познакомиться с поиском записей по образцу.

Порядок выполнения работы

- Откройте программу Access. В окне управления СУБД необходимо включить переключатель **Новая база**. В поле **Имя файла** введите имя файла **is.mdb**
- В окне (рис.12) выберите пункт меню **Создание таблицы в режиме конструктора** (дважды кликнув мышкой на нем).
- В диалоговом окне создания таблицы в поле **Имя поля** вводятся имена полей будущей таблицы БД, а в поле **Тип данных** выбирается соответствующий тип данных. Заполните поля данными из табл. 1.

Таблица 1. Структура таблицы **Наименование**

Имя поля	Тип данных	Общие
Код_документа	Текстовый	Размер поля – 50 Обязательное поле – Да Пустые строки – Нет Индексированное поле – Да (совпадения не допускаются)
Наименование_док	Текстовый	Размер поля – 255 Обязательное поле – Да Пустые строки – Да Индексированное поле – Да (совпадения допускаются)
Текст_док	Поле МЕМО	Обязательное поле – Нет Пустые строки – Да Индексированное поле – Нет

- Определите ключевое поле **Код_документа**. Для этого выделите это поле и нажмите кнопку **Ключевое поле** .
- Закройте таблицу, сохранив ее под именем **Наименование** (рис. 13).
- Заполните базу данных значениями из табл. 2.

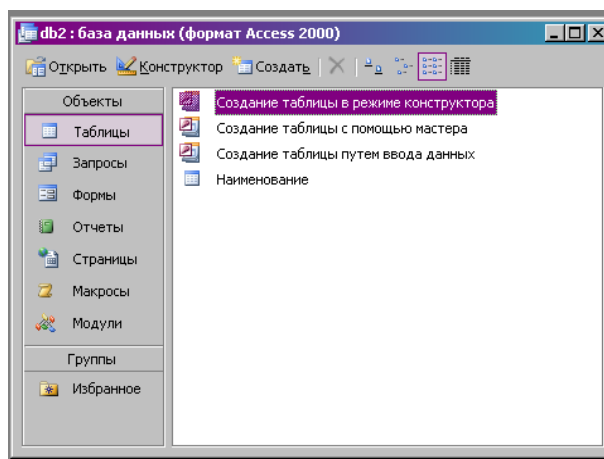


Рис.12. Диалоговое окно управления БД



Рис.13. Сохранение таблицы

Таблица 2. Значения таблицы **Наименование**

Код_док	Наименование_док	Текст
МНС 126-2004	Правила пожежної безпеки в Україні.	
МНС 70-2004	Про затвердження положення про добровільні пожежні дружини та типового положення про пожежно-технічну комісію.	
МНС 59-2006	Про затвердження Інструкції з організації роботи органів державного пожежного нагляду.	
МНС 161-2005	Перелік однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації.	
ППБ-153-90	ППБ для мест рассредоточения эвакуированного населения.	
ППБ-154-90	Правила промышленной безопасности нефтебаз и складов нефтепродуктов.	
ДБН В.2.2-9-99	Громадські будинки та споруди. Основні положення.	
ДБН В.1.1 - 7 – 02	Пожежна безпека об'єктів будівництва.	
ДБН 360-92*.	Містобудування. Планування і забудова міських та сільських поселень.	
ДБН В.2.5-13-98	Пожарная автоматика зданий и сооружений.	
СНиП 2.11.06-91	Склады лесных материалов. Противопожарные нормы проектирования.	
СНиП 2.11.03-93	Склады нефти и нефтепродуктов. Противопожарные нормы.	
СНиП 2.04.02-84*	Водоснабжение. Наружные сети и сооружения.	




- Используя режим конструктор создать таблицы **Реестр документов** и **Тип**.
- Структура таблицы **Реестр документов** представлена в табл. 3.
Структура таблицы **Тип** представлена в табл.4.

Таблица 3. Структура таблицы **Реестр документов**

Имя поля	Тип данных	Общие
Код_документа	Текстовый	Размер поля – 50 Обязательное поле – Да Пустые строки – Нет Индексированное поле – Да (совпадения допускаются)
Тип_док	Текстовый	Размер поля – 50 Обязательное поле – Да Пустые строки – Нет Индексированное поле – Да (совпадения допускаются)
Дата	Дата/Время	Формат поля – Краткий формат

Таблица 4. Структура таблицы **Тип**

Имя поля	Тип данных	Общие
Тип_док	Текстовый	Ключевое поле Размер поля – 50 Пустые строки – Нет Обязательное поле – Да Индексированное поле – Да (совпадения не допускаются)
Наименование_тип	Текстовый	Размер поля – 255 Пустые строки – Да Обязательное поле – Да Индексированное поле – Да (совпадения допускаются)

- Для сортировки записей в таблице используются кнопки  и  сортировки **по возрастанию** и **по убыванию**. Для сортировки необходимо открыть нужную таблицу, установить курсор в любую ячейку нужного столбца и нажать кнопку соответствующей сортировки. Отсортируйте таблицу **Наименование**:
 - столбец **Код_документа** — по возрастанию;
 - столбец **Наименование_док** — по убыванию.
- Для поиска записей по образцу необходимо установить курсор в нужный столбец таблицы, нажать кнопку **Найти** , в открывшемся окне (рис.14) в поле **Образец** ввести текст для поиска, выбрать необходимые

параметры и нажать кнопку **Найти далее**. Выполните поиск всех документов, которые имеют в своем названии символы «СНИП 21»

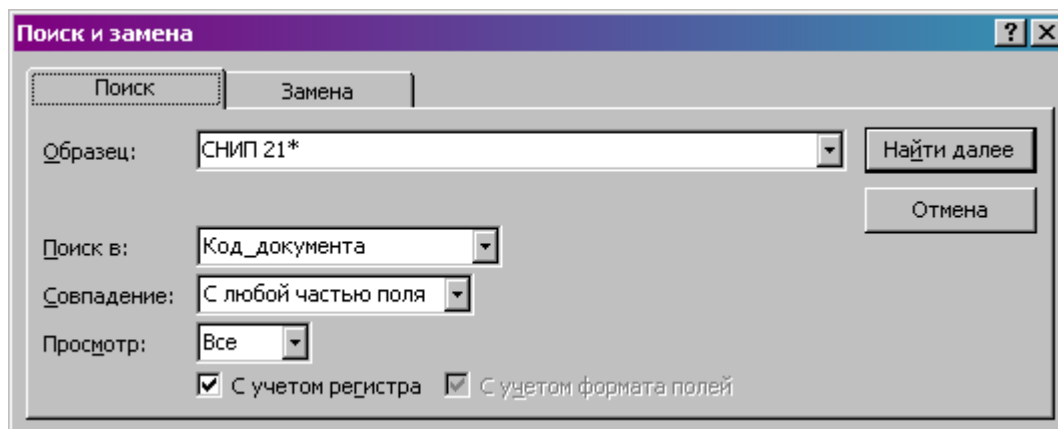


Рис.14. Диалоговое окно **Поиск и замена**

3.3. Создание реляционной БД. Связь таблиц. Добавление и удаление записей в БД. Использование фильтров для поиска информации.

Цели работы:

- научиться создавать реляционные БД;
- научиться создавать связи между таблицами БД;
- познакомиться с поиском записей при помощи фильтра.

Порядок выполнения работы

- Открыть БД is.mdb.
- Открыть таблицу **Тип** (дважды кликнув по имени таблицы) и заполнить ее значениями из табл. 5.

Таблица 5. Значения таблицы **Тип**.

Тип_док	Наименование_тип
ГОСТ	ГОСТы
НПБ	Нормы пожарной безопасности
ППБ	Правила пожарной безопасности
Прочее	Другие документы
РД	Рабочая документация
СНИП	Строительные нормы и правила

- Для создания связи между таблицами необходимо выполнить команду **Сервис-Схема данных**. В появившемся окне необходимо выделить все таблицы и нажать кнопку **Добавить**. Для связывания таблицы **Тип** с таблицей **Реестр** необходимо перетащить ключевое поле таблицы **Тип** (**Тип_док**) на одноименное поле таблицы **Реестр**, при этом появится диалоговое окно **Связи** (рис. 15) в котором необходимо включить флажки **Обеспечение целостности данных**, **каскадное обновление связанных полей**, **каскадное удаление связанных записей** и нажать кнопку **Создать** (или **ОК**). Аналогично устанавливается связь между таблицами **Наименование** и **Реестр**. После создания связей окно **Схема данных** необходимо закрыть, сохранив изменения.
- Заполните таблицу **Реестр** значениями из табл.6. Для удобства работы с таблицей **Реестр** при заполнении можно воспользоваться значениями из таблиц **Тип** и **Наименование**. Для этого откроем таблицу **Реестр** в режиме **Конструктор**. Выделим поле **Тип_док**, связанное с таблицей **Тип**. Выберем вкладку **Подстановка** (рис.16). В поле **Тип элемента управления** выберем **Поле со списком**. В поле **Тип источника строк** – **Таблица или запрос**. В поле **Источник строк** нажмем кнопку **...** После этого в появившемся окне выберем таблицу **Тип** и нажмем кнопку **Добавить**. В окне **Построитель запросов** (рис.17)

перетащим имя поля **Тип_док** на поле **Поле** и закроем окно, сохранив изменения.

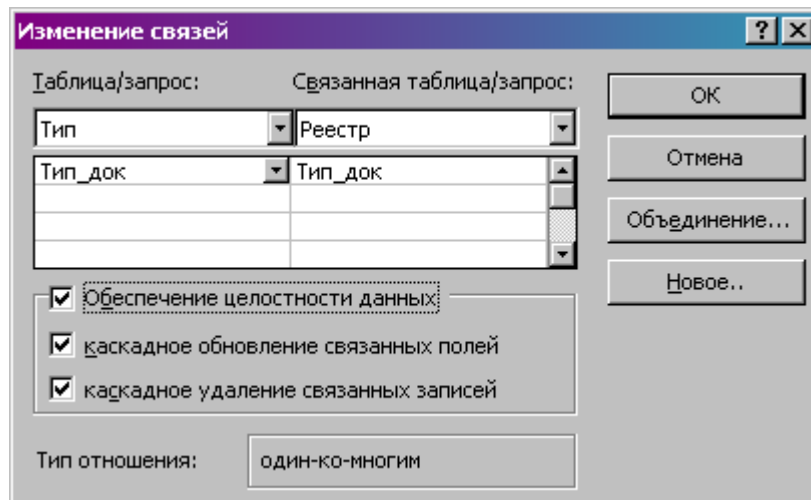


Рис. 15. Окно **Связи**

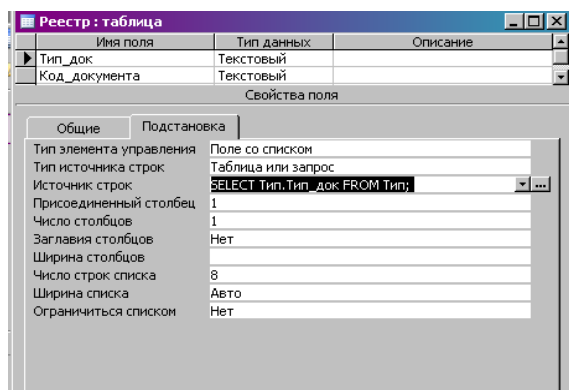


Рис.16. Окно **Конструктор** таблицы **Реестр**

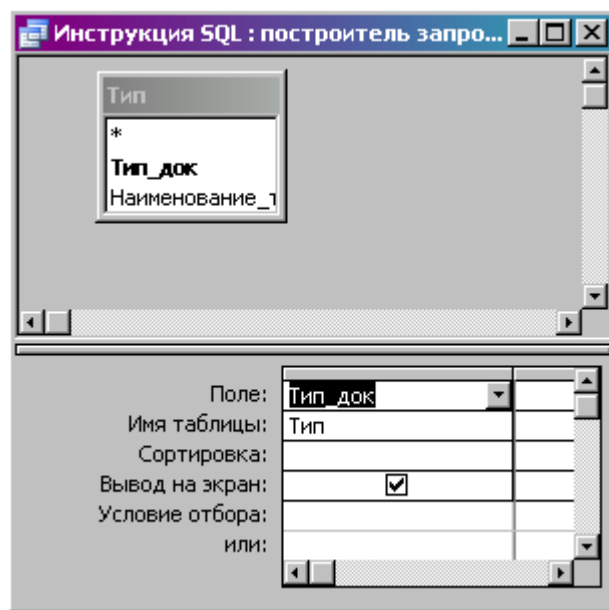


Рис.17. Окно **Построитель запросов**

Таблица 6. Значения таблицы **Реестр**

Тип_док	Код_документа	Дата
МНС	МНС 126-2004	10/19/2004
МНС	МНС 70-2004	02/11/2004
МНС	МНС 59-2006	02/06/2006
МНС	МНС 161-2005	08/22/2005
СНИП	СНиП 2.11.06-91	06/11/1991
СНИП	СНиП 2.11.03-93	03/11/1993
СНИП	СНиП 2.04.02-84*	02/04/1984

- Аналогичным образом подключается ввод значений из таблицы **Наименование**. Связь устанавливается по полю **Код_документа**.
- Для ввода данных в таблицу можно создать специальные формы. Для этого необходимо в окне **Управления БД** (рис.12) выбрать вкладку **Формы** и нажать кнопку **Создать**. В открывшемся окне выбрать **Автоформа в столбец**, выбрать имя таблицы для которой создается форма и нажать **ОК**. Полученная форма представлена на рис.18.

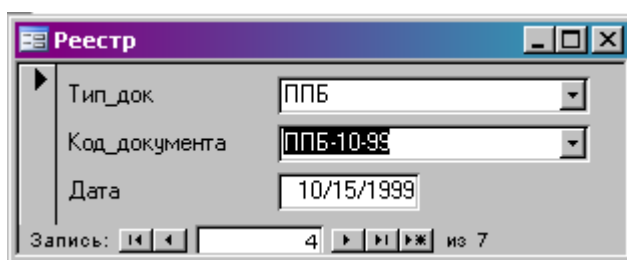





Рис.18. Форма ввода данных для таблицы **Реестр**

- Создайте формы для ввода данных в таблицы **Тип** и **Наименование** и добавьте записи (табл. 7) при помощи форм.

Таблица 7. Значения таблицы **Наименование**

Код_документа	Наименование_док
РД 009-01-96	Установки пожарной автоматики. Правила технического содержания.
РД 25.952-90	Системы автоматические пожаротушения, пожарной, охранной и охранно-пожарной сигнализации. Порядок разработки задания на проектирование.
РД 25.953-90	Системы автоматические пожаротушения, пожарной, охранной и охранно-пожарной сигнализации. Обозначения условные графические элементов связи.
РД 78.145-93	Системы и комплексы охранной, пожарной и охранно-пожарной сигнализации. Правила производства и приемки работ.
Р3112194-0366-97	Нормы расхода топлива и смазочных материалов на автомобильном транспорте

- *Каскадное удаление записей.* Откройте таблицу **Тип** и добавьте в нее запись **Тип_док – Другие, Наименование_тип – Другие**. Сохраните таблицу. Откройте форму для ввода данных в таблицу **Реестр** и добавьте две записи, указав в качестве **Тип_док** имя **Другие**. Откройте таблицу **Тип** и удалите запись **Тип_док – Другие** при помощи контекстного меню. Убедитесь, что в таблице **Реестр** все записи, у которых **Тип_док** был **Другие** удалены.
- *Каскадное обновление записей.* Откройте таблицу **Реестр** и добавьте в нее две записи, выбрав **Тип_док – Прочее**. Откройте таблицу **Тип** и измените **Тип_док – Прочее** на **Другие**. Откройте таблицу **Реестр** и убедитесь в том, что записи обновлены.
- *Поиск записей при помощи фильтра.* Откройте таблицу **Реестр** и установите курсор в поле **Дата**. Нажмите кнопку  **Фильтр по выделенному**. На экран выведутся все записи, которые соответствуют выделенному полю. Для отмены фильтра необходимо нажать кнопку **Удалить фильтр** . Для изменения фильтра необходимо нажать кнопку **Изменить фильтр** . Измените фильтр для поиска всех записей за **12/12/03**. Используя фильтр, найдите все записи, у которых **Тип_док – ППБ**. Найдите все записи, у которых **Код_документа – ППБ***.

3.4. Создание форм ввода-вывода. Создание главной кнопочной формы

Цели работы:

- научиться создавать формы при помощи **Мастера форм** и **Конструктора**;
- научиться создавать кнопочные формы.

Порядок выполнения работы



- Открыть БД is.mdb.
- Перейдите на вкладку **Форма** и нажмите кнопку **Создать**. В открывшемся окне выберите пункт **Конструктор**. Появится пустая форма. На панели элементов (рис.19) выберите элемент **Надпись** . Щелкните мышкой на форме там, где будет размещаться надпись и введите: **Информационная система Нормативные документы**. После чего нажмите **Shift+Enter**. Задайте размер шрифта 14 пт, полужирный и цвет – красный. Если надпись не помещается в выделенное поле, измените его размеры при помощи угловых маркеров. Сохраните форму под именем **Главная**.



Рис. 19. Панель элементов

- Выберите на **Панели элементов** элемент **Кнопка** . Щелкните мышкой в том месте, где должна размещаться кнопка. Появится диалоговое окно **Создание кнопок**.
- Выберите категорию – **Работа с формой**, а действие – **Открытие формы** и нажмите кнопку **Далее**.
- Выберите форму, которая будет открываться при помощи этой кнопки, **Тип** и нажмите **Далее**.
- Задайте имя кнопки (например, **Тип**) или выберите для нее рисунок и нажмите **Готово**.
- Аналогично создайте кнопки для открытия форм **Наименование** и **Реестр**.
- Для выхода из приложения создайте кнопку и выберите категорию **Приложение**, а действие – **Выйти из приложения**. Нажмите **Далее**.
- Выберите рисунок – **Стоп** и нажмите **Готово**.
- Сохраните форму. Выйдите из режима **конструктора**. Откройте форму и проверьте работу соответствующих кнопок.

Самостоятельное задание: Для всех форм ввода добавить кнопки перехода между записями, кнопки выхода из форм и кнопки удаления текущей записи.

3.5. Создание и использование запросов. Создание запросов на выборку

Цели работы:

- закрепить навыки по редактированию таблиц;
- познакомиться с основными видами запросов;
- научиться создавать запросы на выборку различными способами.

Порядок выполнения работы

- Открыть БД is.mdb.
- Создание запросов позволяет извлечь из базы данных информацию, которую необходимо проанализировать или изменить. Данные по запросу выводятся в отдельную таблицу, где их можно редактировать, изменять и т.д. Чаще всего применяется общий тип запроса – запрос на выборку, результат которого отображается в отдельной таблице. Для создания запросов в ACCESS имеется специальный **Мастер** для разработки сложных запросов.
- *Создание и выполнение запросов на выборку.* Для поиска всех записей о Строительных нормах и правилах необходимо:
- Открыть вкладку **Запросы** окна **Управления БД** (рис.12) и нажать кнопку **Создать**, в появившемся окне выбрать **Конструктор** и нажать **ОК**.
- В окне **Добавление таблицы** необходимо выбрать таблицы, данные из которых будут выводиться по запросу. Для этого необходимо выделить

таблицу и нажать кнопку **Добавить**. Добавьте все существующие таблицы.

- В окне **Запрос-запрос на выборку** (рис. 20) в строке **Поле** необходимо выбрать поля, которые будут выведены в результирующей таблице запроса. Если в строке **Вывод на экран** установлен флажок , то поле выводится в таблице, в противном случае не выводится. В строке **Условия отбора** необходимо ввести критерии выбора по конкретному полю. В нашем случае значение поля **Код_документа** должно быть равно «СНИП» (Like «СНИП*»).

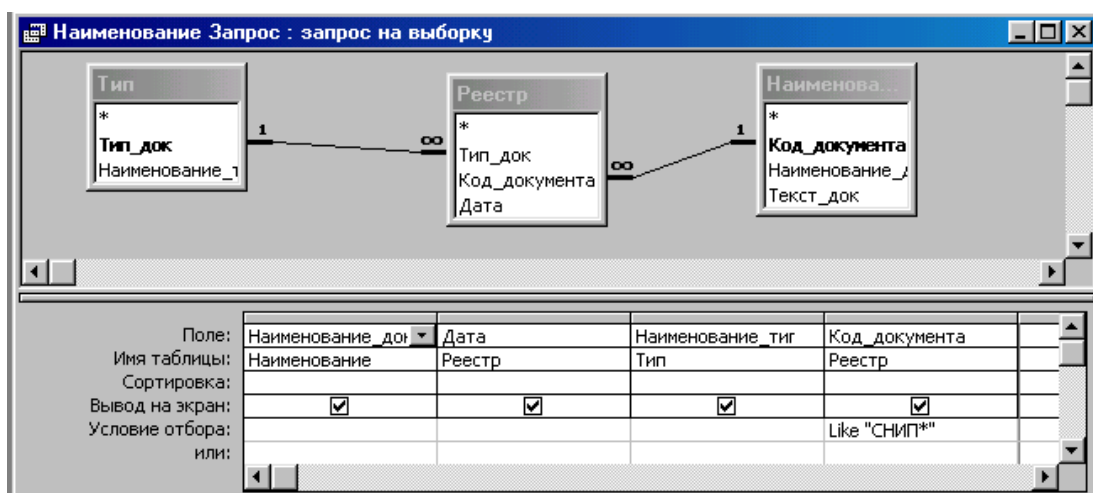


Рис.20. Окно **Запрос – запрос на выборку**


- Для запуска запроса необходимо нажать кнопку **Запуск**  или выполнить команду **Запрос-Запуск**. Сохраните запрос под именем **Запрос_Код** и закройте его. Возможные варианты отбора данных представлены в табл. 8.

Таблица 8. Условия отбора данных

Условия отбора	Результат
<100	Только числа меньше 100
>200	Только числа больше 200
<=75	Числа не больше 75
Between 1/1/03 and 2/28/03	Любой день в январе и феврале 2003 года
2/*/03	Любой день в феврале 2003 года
«СНИП» или «ГОСТ»	Или «СНИП» или «ГОСТ»
Not 20	Все записи, значение которых не равно 20
In (TX, FL)	Записи данного поля, содержащие элементы TX или FL

Самостоятельное задание. Создать запросы на выборку:

- найти все документы, принятые в январе-феврале 2001 года;
- найти все рабочие документы;


- найти все СНИП, которые связаны с нормами и правилами на складах;
 - найти все документы, принятые в марте 2006 года;
 - найти все правила пожарной безопасности для учреждений культуры;
 - найти все СНИП, изданные в 2001 году;
 - найти все ППБ, изданные до 1998 года.
- При построении запросов на выборку можно использовать **Мастер построения запросов**. Для запуска **Мастера** необходимо на вкладке **Запросы** выбрать пункт **Создание запроса с помощью мастера**.

3.6. Создание и использование запросов. Создание запросов на обновление, на добавление, на удаление и на создание таблицы. Создание перекрестных запросов

Цели работы:

- закрепить навыки по созданию таблиц;
- познакомиться с перекрестными запросами;
- научиться создавать запросы на обновление, на добавление, на удаление и на создание таблиц.

Порядок выполнения работы

- *Создание запроса на обновление данных в таблице БД.* Для замены в таблице **Тип Тип_док – СНИП** на **Тип_док – СНИП** необходимо:
 - Перейти на вкладку **Запрос** окна **Управление БД** (рис. 12) и нажать кнопку **Создать**, в появившемся окне выбрать режим **Конструктор** и нажать **ОК**;
 - В окне **Добавление таблицы** необходимо добавить таблицу **Тип**, в меню **Запрос** необходимо выбрать тип запроса **Обновление**;
 - В окне **Запрос на обновление** (рис. 21) в строке **Поле** необходимо выбрать поле которое будет обновляться **Тип_док** (его можно просто перетащить из таблицы **Тип** на нужную строку). В строке **Обновление** необходимо указать необходимое выражение или значение (**СНИП**), если необходимо, то можно указать условия отбора.
 - Сохраните запрос, дав ему, имя **обновление** и выполните его, нажав кнопку **Запуск** . Просмотрите полученные изменения в таблицах **Тип** и **Реестр**.

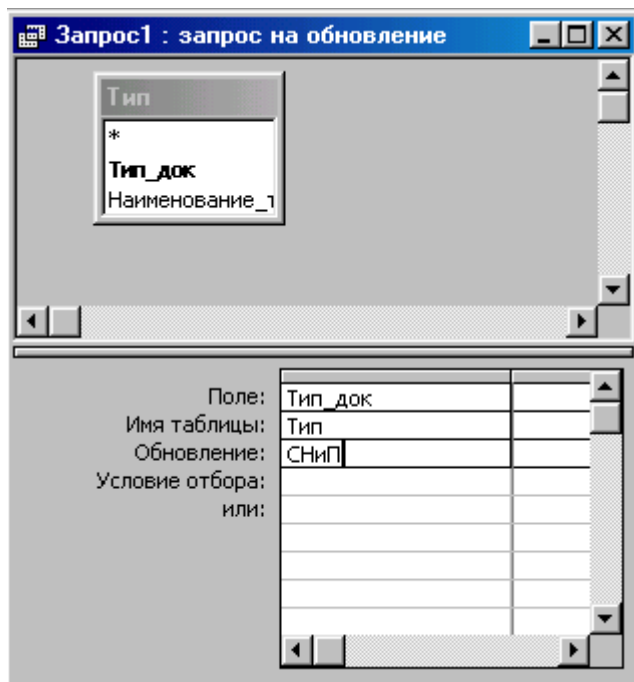


Рис. 21. Окно **Запрос на обновление**

- **Запрос на добавление** используется для вставки группы записей из одной таблицы в другую, причем таблица может находиться в другой БД. Для этого необходимо:
 - создайте копию БД is.mdb, присвоив ей имя is2.mdb;
 - перейдите на вкладку **Запрос** и нажмите кнопку **Конструктор**, в окне **Добавление таблицы** выберите и добавьте таблицу **Реестр**;
 - выполните команду **Запрос-Добавление** для выбора типа запроса;
 - В окне **Добавление** (рис. 22) необходимо указать имя БД и имя таблицы в которую добавляются записи (**имя файла – is2.mdb**, **имя таблицы – Реестр**)

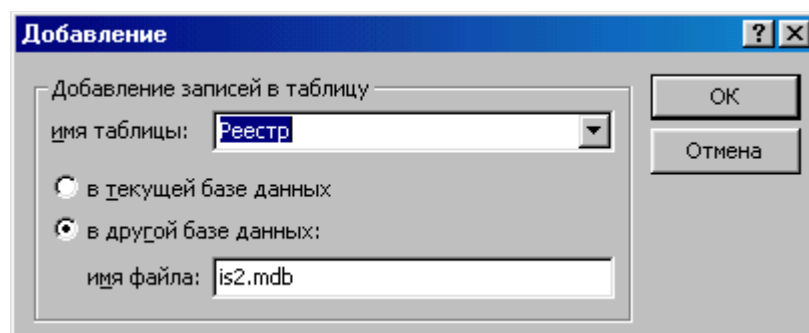


Рис. 22. Окно **Добавление**

- Сохраните запрос, дав ему имя **добавление**;
- Откройте БД is2.mdb для установления связи с БД is.mdb. При открытии is2.mdb база данных is.mdb будет закрыта, поэтому необходимо сохранить все изменения в ней. Закройте БД is2.mdb и

откройте is.mdb. Выберите вкладку **Запросы**, выберите запрос **добавление** и нажмите кнопку **Конструктор**.

- В окне **Запрос на добавление** (рис. 23), в строке **Поле** необходимо указать имена тех полей значения, которых будут добавлены. Дополнительно можно указать условия отбора. Сохраните запрос и выполните его. Откройте БД is2.mdb и просмотрите полученные изменения.

Самостоятельное задание. Добавьте в БД is2.mdb из таблицы **Реестр** только те записи, которые были внесены в 2007 году.

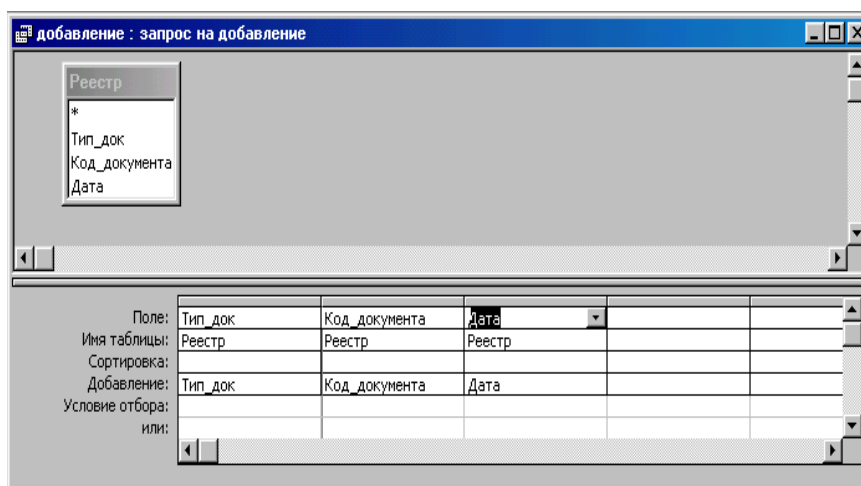


Рис. 23. Окно **Запрос на добавление**

- **Запрос на удаление** - это активный запрос, позволяющий удалить записи в выбранной таблице. Например, для удаления всех записей из БД is2.mdb таблицы **Реестр** за 2003 год необходимо:
 - Открыть БД is2.mdb. Открыть таблицу **Реестр**, просмотреть ее содержимое (если нет записей за 2003 год, создайте их или исправьте существующие).
 - Перейти на вкладку **Запросы** и нажать кнопку **Конструктор**, в окне **Добавление таблицы** выбрать таблицу из которой удаляются записи – **Реестр** и нажать кнопку **Добавить**.
 - В окне **Запрос на удаление** (рис. 24) из списка полей таблицы **Реестр** необходимо перетащить (*) в бланк запроса в строку **Поле**. В ячейках **Удаление** вводятся слова **Из** и **Условие**. Первое относится к таблице, откуда удаляются записи, а второе – к полям, для которых задаются условия отбора. Для удаления записей за 2003 год в строке **Условие отбора** необходимо ввести **Between #01.01.03# And #31.12.03#**, что позволяет удалить все записи за 2003 год.
 - Сохраните запрос и выполните его. Просмотрите изменения в таблице **Реестр**.

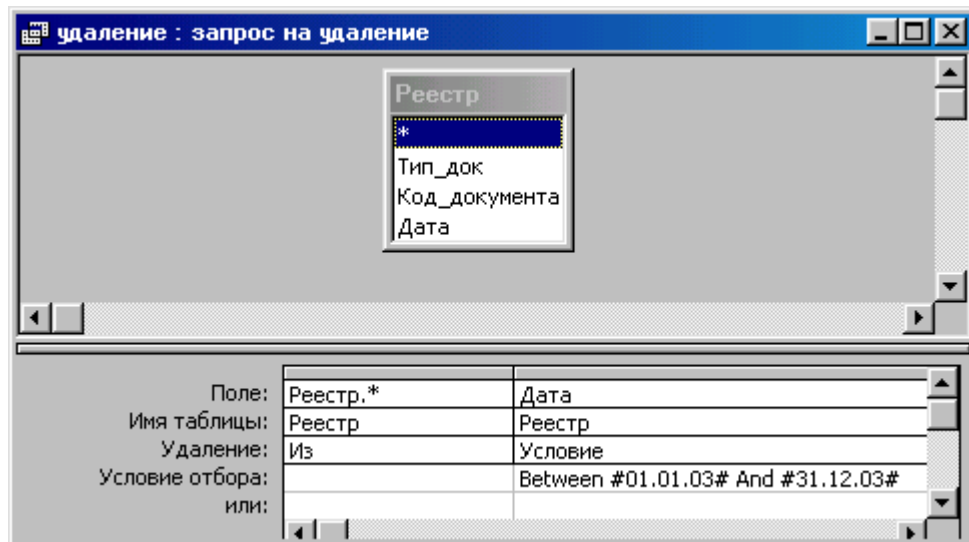


Рис. 24. Окно **Запрос на удаление**

3.7. Создание отчетов

Благодаря использованию отчетов можно распечатать документы из БД в удобной для пользователя форме. Источником данных служат таблица или запрос. В Access предусмотрены разнообразные способы форматирования отчетов.

3.7.1. Создание отчетов при помощи команды Автоотчет

Наиболее быстрый способ создания отчета – использование одного из автоотчетов. После просмотра результата можно изменить выбранный шаблон в режиме Конструктора. Для подготовки автоотчета необходимо:

- Открыть вкладку **Отчеты** в окне управления БД (рис.12) и нажать кнопку **Создать**.
- В диалоговом окне **Новый отчет** выберите пункт **Автоотчет в столбец** и указать источник данных – **таблицу** или **запрос**. В нашем случае выбрать запрос – **Запрос_Код** и нажать **ОК**.

3.7.2. Использование Мастера отчетов

- Открыть вкладку **Отчеты** в окне управления БД (рис.12) и нажать кнопку **Создать**.
- В диалоговом окне **Новый отчет** выберите пункт **Мастер отчетов** и нажмите **ОК**.
- В первом окне **Мастера** укажите таблицу или запрос, которые служат источником данных для отчета, а также отображаемые в отчете поля (таблицы – **Наименование** и **Тип** и все поля таблиц) и нажмите кнопку **Далее**.

- В окне **Добавить уровни группировки** необходимо выбрать поле по которому будет выполняться группировка записей – **Тип_документа** и нажать **Далее**.
- В окне **порядок сортировки** выбирается поле по которому выполняется сортировка и параметры сортировки и нажимается кнопка **Далее**.
- В следующем окне выбрать приемлемый макет будущего отчета — например, **ступенчатый** и нажать **Далее**.
- В окне **выберите требуемый стиль** определите стиль отчета — например, **деловой** и нажмите **Далее**.
- Задайте имя отчета **Отчет_Документы** и нажмите **Готово**.

РАЗДЕЛ 4. СОСТАВ И СТРУКТУРА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Главной целью работы является закрепления знаний, умений и навыков, полученных в процессе изучения курса «АСУ и информационное обеспечение деятельности службы гражданской защиты» путем самостоятельного решения поставленной задачи.

Задачей работы является проектирование и компьютерная реализация упрощенной информационной системы в области будущей профессиональной деятельности в подразделениях МЧС (приложение 1).

Работа выполняется слушателями в бригадах из 2-3 человек, разрабатывается ими самостоятельно и подается каждым слушателем к защите в виде личного отчета (15-20 страниц). Отчет представляется в электронном варианте (на дискете) и напечатанном (прошитом и в «файле»).

Требования к тексту отчета:

1. Перед введением текста рекомендуется установить следующие параметры страницы: верхнее, нижнее, левое и правое поля — 2 см, 2 см, 2,5 см и 1,5 см соответственно; от края до нижнего колонтитула — 1,1 см, до верхнего — 1,25 см.
2. Текст отчета набирается шрифтом типа Times New Roman и размером для основного текста — 14 пт, для заголовков — 14 пт полужирным, размещается по ширине страницы, абзацный отступ первой строки — 1,25 см, междустрочный интервал — полуторный.
3. Желательно выделить некоторые слова полужирным шрифтом, курсивом; вставить в текст изображения окон и кнопок, которые использовались при создании таблиц и диаграмм.
4. Номера страниц вставить внизу страницы от центра.

Отчет должен включать в себя:

1. Титульный лист.
2. Постановку задачи.

3. Описание проекта базы данных.
 - 3.1. Перечень информационных задач.
 - 3.2. Описание каждой задачи (бланк задачи).
 - 3.3. Описание каждого информационного объекта (бланки объектов)
 - 3.4. Схему логических связей объектов.
4. Базу данных на магнитном носителе. Для базы должны быть созданы формы для ввода/вывода информации в таблицы, а также запросы на выборку данных из таблиц. Для просмотра запросов должны быть созданы отчеты. Примеры заполненных таблиц, а также результаты выполнения запросов также должны быть в отчете.
5. Подробное описание всех действий по выполнению самостоятельной работы в среде MS Access — создание базы данных, ее заполнение, установление логических связей между таблицами, создание форм для введения, удаления, редактирования записей в таблицах, создание запросов, создание отчетов.

ВНИМАНИЕ! Отчет, который будет оформлен с нарушением этих требований, к защите не принимается!

Титульный лист заполняется в соответствии с приложением 2.

При **постановке задачи** слушатель формулирует цели и задачи контрольной работы, а также описывает предметную область, приводит перечень проанализированных входных документов и те сведения, которые пользователь хочет получить от информационной системы.

Описание проекта базы данных, который включает в себя перечень информационных задач, бланки задач, информационных объектов и схему связей выполняется в соответствии с примером, рассмотренным во 2 разделе методических указаний.

Файлы базы данных заполняются средствами СУБД MS Access в соответствии с примером, рассмотренным в разделе 3. Каждая таблица должна содержать не менее 20 записей.

Описание всех действий по компьютерной реализации проекта должно обязательно присутствовать в отчете.

Список литературы

1. Дж. Ульман. Основы систем баз данных. — М.: Финансы и статистика, 1983. — 150 с.
2. Николаева О.Г. Методические указания для выполнения курсовой и лабораторных работ по курсу «Вычислительная техника и программирование». — Харьков: ХГТУСА, 1999. — 47 с.
3. Золотова С.И. Практикум по Access. — М.: Финансы и статистика, 2000. — 144 с.
4. Ю. Стоцкий. Самоучитель Office2000. — Спб.: Питер, 2002. — 608 с.
5. Стив Сагман. Microsoft Office 2000. — М.: ДМК Пресс, 2002. — 672 с.

ПРИЛОЖЕНИЕ 1. ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ

1. Автоматизированная информационная справочная система «Право». ИС позволяет в автоматизированном режиме вести и актуализировать информационный массив правовых документов, сократить время обработки и получения справочной информации по вопросам правового регулирования в области пожарной безопасности и деятельности подразделений МЧС.

2. Информационно-поисковая система по пожаровзрывоопасности веществ и материалов и средствам их тушения. Сведения:

- названия и синонимы названий;
- общие характеристики;
- химические формулы;
- показатели пожаровзрывоопасности;
- физико-химические показатели;
- несовместимость;
- средства тушения.

3. Автоматизированная информационная система «Средства обеспечения пожарной безопасности». БД содержит сведения о средствах обеспечения пожарной безопасности:

- технико-экономические и тактико-технические характеристики;
- изображения внешнего вида изделий;
- сведения о сертификации и др.

4. Автоматизированная информационная система «Средства обеспечения ведения аварийно-спасательных работ». БД содержит сведения о фирмах, изготавливающих и поставляющих пожарно-техническую продукцию:

- коды ОКПО;
- наименования: полное и краткое;
- номера и сроки действия лицензий;
- адреса: почтовый, E-mail, Internet;
- основные руководители и подразделения;
- телефоны, факсы, телетайпы;
- номенклатура продукции и др.

5. Информационно-поисковая система «Совместимость веществ и материалов». Содержит сведения об опасном взаимодействии веществ и материалов; сведения о веществах. Для каждого вещества указаны:

- наименование;
- перечень несовместимых веществ;
- вид опасного взаимодействия;

- источники информации.

6. Информационная система «АРМ-Лицензирование». Предусматривает полный цикл обработки документов, ведение реестра выданных лицензий и получение отчетных справок по выданным лицензиям.

7. Информационная система для инспектора ГПН на объекте (на производстве, метрополитене и т.п.). Автоматизация операций инспектора ГПН по вводу, обработке, анализу нормативных документов и сведений об объекте (зданиях, цехах, депо, подвижном составе и т.п.), а также учету и контролю административных дел и предписаний ГПН.

8. Программное средство «Приостановка эксплуатации объектов». Информационная поддержка инспектора ГПН при принятии решения о приостановке эксплуатации различных объектов на основе анализа нормативных документов, автоматизации составления отчетной документации и контроля графиков проверок объектов.

9. Программное средство «Музей». Обработка и обобщение данных о фактическом уровне пожарной безопасности музейных объектов, определение достаточности противопожарных мероприятий, установление эффективных путей снижения пожарного риска и контроля за пожарной безопасностью конкретных музеев.

10. Программное средство «Автоматизированная система по информационному обеспечению РТП при тушении пожаров в резервуарах». Содержит и выдает в диалоговом режиме информацию по основным типам резервуаров, нефти и нефтепродуктам, отечественным и импортным пенообразователям. Осуществляет расчет сил и средств в зависимости от вида продукта и огнетушащего вещества, в том числе при тушении после длительного свободного горения и при наличии усложняющих тушение факторов.

11. Программное средство «АРМ Пожарно-прикладной спорт». Учет и анализ результатов соревнований по пожарно-прикладным видам спорта.

12. Программное средство «АРМ-Техника». Информационная поддержка специалиста отдела техники ГПС (региональный уровень) на основе учета и анализа данных об эксплуатации пожарно-спасательной техники и вооружения.

13. Программное средство «АРМ-Кадры». Автоматизация решения задач в области кадрового обеспечения, таких, как: формирование и ведение банка данных по личному составу органов управления и подразделений МЧС; обеспечение работы специалистов в режиме «запрос-ответ»;

обоснование управленческих решений; внедрение элементов безбумажной информационной технологии в подразделениях МЧС; обеспечение информационного взаимодействия с банком данных по кадровому составу МЧС; получение графиков и таблиц на основе выбранных статистических показателей.

14. Информационно-поисковая система «Пресса». Формирование БД по публикациям в печатных СМИ на пожарно-спасательную тематику. Получение аналитических данных в табличном виде для информационной поддержки принятия управленческих решений по взаимодействию со СМИ.

15. Программное средство «АРМ инженера ИПЛ». Автоматизация таких видов деятельности инженера ИПЛ, как:

- ведение банка данных по исследованным пожарам;
- ведение банка данных по результатам исследования предметов, изъятых с места пожара;
- подготовка и оформление технических заключений по причинам пожара и результатам исследования вещественных доказательств;
- подготовка отчетных материалов.

16. Программное средство «АРМ-Склад». Автоматизация решения задач по вещевому обеспечению личного состава МЧС.

17. Программное средство по автоматизации рабочего места диспетчера гарнизона МЧС и пожарно-спасательных частей «Диспетчер оперативно-диспетчерской службы». Автоматизация решения основных задач диспетчерами пожарно-спасательных частей. Представляет собой интегрированный программный продукт с возможностью дальнейшего расширения процедур и состава задач.

18. Информационно-поисковая система «Банк идентификация». Решение задач идентификации и установления соответствия применяемого материала или средства огнезащиты требованиям, приведенным в нормативно-технической документации. Содержит термоаналитические данные и характеристики, определяющие пожарную опасность веществ и материалов и эффективность антипиренов и средств огнезащиты, а также данные, применяемые для идентификации образцов.

19. Автоматизированная информационно-поисковая база «Библиотека ПБ». Автоматизированная информационно-поисковая база полных текстов нормативных документов в области пожарной безопасности.

ПРИЛОЖЕНИЕ 2. ТИТУЛЬНЫЙ ЛИСТ КОНТРОЛЬНОЙ РАБОТЫ

УНИВЕРСИТЕТ ГРАЖДАНСКОЙ ЗАЩИТЫ УКРАИНЫ

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И СИСТЕМ УПРАВЛЕНИЯ**

ОТЧЕТ ПО КОНТРОЛЬНОЙ РАБОТЕ ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ СРЕДСТВАМИ СУБД MS ACCESS

Тема: Автоматизированная информационная справочная система «Право»

Выполнил слушатель группы М-07

капитан вн.сл. Шевченко Т.Г.

Проверил доцент кафедры ИТиСУ

к.т.н Шматко О.В.

Оценка: _____

Дата: _____