

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)**

**РАБОТА С ЦЕНТРАЛИЗОВАННЫМИ
И РАСПРЕДЕЛЕННЫМИ БАЗАМИ ДАННЫХ**

Самара 2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

РАБОТА С ЦЕНТРАЛИЗОВАННЫМИ И РАСПРЕДЕЛЕННЫМИ БАЗАМИ ДАННЫХ

Составители *О.К. Либерзон, Е.И. Чигарина*

САМАРА

Издательство Самарского университета

2017

УДК 004(075)

ББК 32.97я7

Составители О.К. Либерзон , Е.И. Чигарина

Рецензент: канд. техн. наук, доц. Л.А. Жаринова

Работа с централизованными и распределенными базами данных: [Электронный ресурс]: метод. указания / сост. *О.К. Либерзон, Е.И. Чигарина*. – Самара: Изд-во Самарского университета, 2017. – 62 с. : ил. Электрон. текстовые и граф. дан. (Кбайт).- 1 эл. опт. диск (CD-ROM)

Методические указания содержат теоретические сведения по организации и реализации централизованных и распределенных баз данных, а также варианты заданий для выполнения лабораторных и самостоятельных работ.

Методические указания предназначены для студентов направления 09.03.01 – «Информатика и вычислительная техника» в качестве методических указаний к лабораторному практикуму по курсу «Базы данных».

Подготовлены на кафедре информационных систем и технологий.

УДК 004(075)

ББК 32.97я7

© Самарский университет, 2017

Оглавление

| | |
|---|----|
| Введение..... | 5 |
| ЛАБОРАТОРНАЯ РАБОТА № 1..... | 6 |
| СОЗДАНИЕ ОТДЕЛЬНЫХ ТАБЛИЦ И СРЕДСТВА ИХ ВЕДЕНИЯ, ИНДЕКСИРОВАНИЕ ТАБЛИЦ, СОЗДАНИЕ БАЗЫ ДАННЫХ..... | 6 |
| ЛАБОРАТОРНАЯ РАБОТА № 2..... | 22 |
| КОМАНДЫ ПЕРЕМЕЩЕНИЯ В ТАБЛИЦЕ, ПОИСК ЗАПИСЕЙ В НЕУПОРЯДОЧЕННОЙ ТАБЛИЦЕ И ОРГАНИЗАЦИЯ ЗАПРОСОВ..... | 22 |
| ЛАБОРАТОРНАЯ РАБОТА № 3..... | 32 |
| СОЗДАНИЕ ОТЧЕТОВ..... | 32 |
| ЛАБОРАТОРНАЯ РАБОТА № 4..... | 37 |
| РАЗРАБОТКА ЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ СРЕДСТВАМИ ERWIN. НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ..... | 37 |
| ЛАБОРАТОРНАЯ РАБОТА № 5..... | 38 |
| РАЗРАБОТКА ФИЗИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ НА ЯЗЫКЕ TRANSACT SQL, КОМАНДЫ ВЕДЕНИЯ ДАННЫХ..... | 38 |
| ЛАБОРАТОРНАЯ РАБОТА № 6..... | 38 |
| РЕАЛИЗАЦИЯ ЗАПРОСОВ НА ЯЗЫКЕ TRANSACT SQL..... | 38 |
| ЛАБОРАТОРНАЯ РАБОТА № 7..... | 39 |
| РАБОТА С КУРСОРАМИ, ВИДАМИ, ПОЛЬЗОВАТЕЛЬСКИМИ ТИПАМИ, ПРАВИЛАМИ,ЗНАЧЕНИЯМИ ПО УМОЛЧАНИЮ НА ЯЗЫКЕ TRANSACT SQL..... | 39 |
| ЛАБОРАТОРНАЯ РАБОТА № 8..... | 40 |
| РАБОТА С ХРАНИМЫМИ ПРОЦЕДУРАМИ И ФУНКЦИЯМИ НА ЯЗЫКЕ TRANSACT SQL..... | 40 |
| ЛАБОРАТОРНАЯ РАБОТА № 9..... | 40 |
| РАБОТА С ТРИГГЕРАМИ НА ЯЗЫКЕ TRANSACT SQL..... | 40 |
| Приложение 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ СИСТЕМЫ VISUAL FOXPRO 9.0 и MICROSOFT OFFICE ACCESS 2010..... | 41 |
| Приложение 2. ОСНОВНЫЕ ТИПЫ ПОЛЕЙ VISUAL FOXPRO 9.0..... | 46 |
| Приложение 3. ОСНОВНЫЕ ТИПЫ ДАННЫХ MICROSOFT ACCESS 2010..... | 48 |
| Приложение 4. ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ..... | 50 |

Введение

База данных – это совокупность структурированных данных, описывающих некоторую предметную область и хранящихся во внешней памяти. Централизованная база данных (БД) - это база данных, хранящаяся в памяти одного компьютера. Распределенная база данных - это база данных, состоящая из одной или нескольких частей, хранящихся в памяти компьютеров вычислительной сети. В зависимости от структуры данных базы данных могут быть иерархическими, реляционными, сетевыми, объектными и другими.

В лабораторном практикуме рассмотрена работа с реляционными базами данных, используя средства их реализации на примере систем управления базами данных (СУБД) Visual FoxPro 9.0, Microsoft Office Access 2010 (как средства реализации централизованных баз данных) и MS SQL Server (как средство реализации распределенных баз данных).

Реляционная модель данных была предложена американским математиком Э. Ф. Коддом в 1970 году в связи с решением следующей задачи – обеспечить независимость представления и описания данных от прикладных программ.

В основе реляционной модели данных лежит понятие отношения (англ. relation).

N-арным отношением R называют подмножество декартова произведения $D_1 \times D_2 \times \dots \times D_n$ различных множеств D_1, D_2, \dots, D_n . Исходные множества D_1, D_2, \dots, D_n называют в модели доменами:

$R \subseteq D_1 \times D_2 \times \dots \times D_n$, где $D_1 \times D_2 \times \dots \times D_n$ – полное декартово произведение. Полное декартово произведение – это набор всевозможных сочетаний из n элементов каждое, где каждый элемент берется из своего домена.

Схемой отношения R является перечень имен атрибутов данного отношения с указанием домена, к которому они относятся:

$$S_R = (A_1, A_2, \dots, A_n), A_i \in D_i$$

Столбцы отношения называют атрибутами и присваивают им имена. Реляционная база данных – это набор взаимосвязанных отношений.

Системы управления базами данных (СУБД) Visual FoxPro 9.0 и Microsoft Office Access 2010 или просто Access, разработанные корпорацией Microsoft, являются в настоящее время одними из наиболее популярных средств реализации баз данных реляционного типа и полностью соответствуют современным требованиям, предъявляемым к системам обработки информации.

Visual FoxPro имеет событийно-управляемый объектно-ориентированный язык программирования и позволяет решать многие задачи визуально.

Access имеет широкий спектр функций, таких как связанные запросы, сортировка по разным полям, связь с внешними таблицами и базами данных, а также встроенный язык Visual Basic for Applications (VBA). С помощью

Access можно создавать приложения, работающие с базами данных. Одновременно Access может работать только с одной базой данных, но одна база данных Access может включать сотни таблиц, форм, запросов, отчетов, макросов и модулей, которые хранятся в одном файле с расширением accdb для Access 2007-2010 или с расширением mdb для предыдущих версий.

Основные характеристики системы VISUAL FOXPRO 9.0 и MICROSOFT OFFICE ACCESS 2010 приведены в приложении 1. Обратите внимание, что некоторые характеристики могут быть ограничены доступной памятью и дисковым пространством.

ЛАБОРАТОРНАЯ РАБОТА № 1

СОЗДАНИЕ ОТДЕЛЬНЫХ ТАБЛИЦ И СРЕДСТВА ИХ ВЕДЕНИЯ, ИНДЕКСИРОВАНИЕ ТАБЛИЦ, СОЗДАНИЕ БАЗЫ ДАННЫХ

Создание таблиц. Теоретические сведения

Создание таблиц в Visual FoxPro 9.0

В Visual FoxPro 9.0 можно создавать как таблицы, входящие в базу данных, так и отдельные таблицы, называемые свободными (free) таблицами. В этой лабораторной работе будут рассмотрены вопросы, связанные с созданием свободных таблиц.

Каждая таблица хранится в памяти на диске в виде файла с расширением .DBF (data base file), например, stud.dbf. Имена таблицы и файла с расширением .DBF совпадают, поэтому, при задании имени таблицы, необходимо учитывать ограничения на имя файла, которые определяются требованиями операционной системы.

Для создания таблицы необходимо определить, на каком диске, и в каком каталоге будет храниться создаваемая таблица.

Для получения информации об активном диске, а также каталоге, в командном окне надо выполнить команду вывода «?» и одну из следующих функций:

| | |
|----------------|--|
| CURDIR(<диск>) | Возвращает прописными буквами имя текущего каталога на активном или заданном диске |
| SYS(5) | Возвращает имя активного диска |
| SYS(2003) | Возвращает имя текущего каталога на диске |
| SYS(2004) | Возвращает полное имя системного каталога |

Установить текущий или активный каталог по умолчанию можно с помощью команды Set Default to <путь>, заданной в командном окне, или при помощи команды системного меню Tools/Options/File Locations/Default Dir...

Создание таблицы заключается в определении структуры таблицы и её заполнении. В свою очередь структура таблицы включает перечень имён полей таблицы (Name) с указанием типов (Type) и размерностей полей (Width – общая длина поля, Decimal – длина дробной части числа), признака пустого поля (Null), а также указание ключевых полей (Index).

Имена полей задаются как прописными, так и строчными буквами. Длина имени поля свободной таблицы не должна превышать 10 символов. Длина имени поля таблицы, входящей в базу данных, может находиться в диапазоне от 1 до 254 символов, при этом использование пробелов в имени не допустимо. Основные типы полей Visual FoxPro 9.0. приведены в приложении 2.

В Visual FoxPro 9.0 существуют следующие способы создания таблицы:

— *Создание таблицы из текста программы.*
Для этой цели предназначена команда SQL CREATE TABLE | DBF.

— *Создание таблицы с помощью командного окна.*

Для создания таблицы в текущем каталоге, используется команда CREATE, при выполнении которой сначала открывается диалоговое окно Create, а затем открывается диалоговое окно Table Designer.

— *Создание таблицы с помощью конструктора.*

Для создания таблицы необходимо выполнить команду системного меню File/New/Table, ввести имя таблицы в открывшемся диалоговом окне Create.

После присвоения имени таблицы открывается окно Table Designer, которое имеет три вкладки: Fields, Indexes, Table.

Для задания структуры таблицы используется вкладка Fields.

— *Создание таблицы с помощью мастера.*

Для создания таблицы необходимо выполнить команду системного меню Tools/Wizard/Table.

В Visual FoxPro существует два режима просмотра таблицы: Edit и Browse.

По умолчанию после создания структуры таблицы устанавливается режим Edit. В этом режиме после ввода информации во все поля одной записи Visual FoxPro автоматически добавляет следующую запись.

В режиме Browse поля размещены в один ряд. Одна строка соответствует одной записи таблицы, а записи размещены одна под другой. При вводе данных в этом режиме после ввода одной записи пользователю необходимо самому добавлять новую пустую запись, используя для этого команду Table/Append New Record или комбинацию клавиш <Ctrl> + <Y>.

Создание таблиц в Microsoft Access 2010

Для создания файла базы данных необходимо выполнить команду меню Файл/Создать/Новая база данных. Далее именуем базу данных,

определяем папку, в которой будет размещен этот файл и щелкаем кнопкой мыши на значке «Создать».

Для создания таблиц в Access существуют следующие основные способы:

- *Режим таблицы.*
- *Конструктор.*
- *Мастер таблиц.*
- *Импорт таблиц.*

Создание таблицы через режим таблицы проходит путем непосредственного ввода данных в таблицу. При сохранении таблицы Microsoft Access проанализирует данные и автоматически присвоит каждому полю соответствующий тип данных и формат. Создание таблиц в режиме таблицы имеет ограниченное применение. Поля таблицы не имеют содержательных имен, так что почти всегда нужно будет редактировать структуру, чтобы переименовать поля. Кроме того, из примера Access не всегда может правильно определить тип данных. Таблицы, которые созданы в режиме таблицы, не могут включать объекты OLE и мемо-поля. Если есть желание иметь такие поля, вам нужно изменить структуру таблицы.

Для создания таблицы через режим таблицы необходимо:

1. Выполнить команду меню **Создание/Таблица**
2. Переименовать нужные столбцы
3. Ввести данные в таблицу
4. Сохранить данные (нажать кнопку **«Сохранить»** или **Файл/Сохранить** или знак **«Закреть таблицу»**)
5. Создать первичный ключ.

В случае необходимости дополнительные столбцы можно добавлять в любое время.

Создание таблицы в режиме конструктора позволяет добавлять поля, настраивать отображение полей и обработку в них данных, а затем создавать первичный ключ.

Для создания таблицы в режиме конструктора необходимо:

1. Выполнить команду меню **Создание/Конструктор таблиц**
2. Определить все нужные поля в таблице. Для этого:
 - 2.1. Щелкните ячейку в столбце **Имя поля** и введите имя поля.

При задании имен полей в таблицах необходимо придерживаться ряда правил:

- Имена полей в таблице не должны повторяться, т. е. должны быть уникальными.
- Имена полей могут содержать не более 64 символов, включая пробелы.
- Желательно избегать употребления имен полей, совпадающих с именами встроенных функций или свойств Microsoft Access (например, Name — имя).

- Имя поля не должно начинаться с пробела или управляющего символа (коды ASCII 00-31).
- Имена полей могут содержать любые символы, включая буквы, цифры, пробелы, специальные символы, за исключением точки (.), восклицательного знака (!), апострофа (') и квадратных скобок ([], []).
- Рекомендуется не использовать очень длинные имена, т. к. они плохо запоминаются, и не включать в имена большое количество пробелов, особенно если с именами предполагается работать с помощью процедур Visual Basic for Application. Чтобы следовать этой рекомендации и при этом получать понятные имена, можно либо использовать вместо пробелов между словами знак подчеркивания (_), либо писать все слова слитно, но каждое начинать с заглавной буквы.

Эти же правила действительны и для имен таблиц, а также других объектов Microsoft Access.

2.2. В столбце **Тип данных** выбрать из раскрывающегося списка тип данных.

Основные типы данных Microsoft Access приведены в приложении 3.

В поле типа можно также выбрать значение **Мастер подстановок** (Lookup Wizard), который запускает Мастера подстановок (Lookup Wizard), создающего поле подстановок. Поле подстановок позволяет выбирать значение поля из списка, содержащего набор постоянных значений или значений из другой таблицы.

2.3. Задать свойства поля.

Список свойств поля зависит от типа поля и отображается, как только выбран тип данных для поля на нижней панели окна. Для того чтобы просмотреть или изменить свойства конкретного поля таблицы необходимо:

- Выделить нужное поле таблицы (то есть установить курсор в строку с именем поля) в окне Конструктора.
- Раскрыть вкладку **Общие** (General) для просмотра общих свойств поля или вкладку **Подстановка** (Lookup) для просмотра параметров подстановки.
- Ввести описание данных в столбце **Описание**, которые будет содержать это поле.

Текст описания будет выводиться в строке состояния при добавлении данных в поле, а также будет включен в описание объекта таблицы. Вводить описание не обязательно.

3. Создать первичный ключ.

Чтобы установить для таблицы первичный ключ и завершить ее создание в режиме конструктора, необходимо выполнить следующие действия:

- 3.1. В режиме конструктора выделить поле, которое будет играть роль первичного ключа.
- 3.2. Кликнуть по кнопке **Ключевое поле** панели инструментов **Конструктора таблиц** или выбрать команду главного меню **Правка – Ключевое поле** (слева возле имени выделенного поля появится символ ключа).
- 3.3. После указания ключевого поля таблицу необходимо сохранить, для чего необходимо кликнуть по кнопке **Сохранить** панели инструментов конструктор таблиц и в открывшемся окне ввести имя таблицы и кликнуть по кнопке **ОК**.

Если первичный ключ не определен, то при выходе из режима конструктора появится соответствующее предупреждение, и программа Access предложит Вам создать ключевое поле до закрытия таблицы.

Основные правила, которые приняты для ключей в Access:

- для удобства ключевое поле обычно указывается в структуре таблицы первым;
 - если для таблицы определен первичный ключ, программа Access автоматически блокирует ввод в это поле повторяющихся значений или значения Null (пусто);
 - Access автоматически сортирует записи таблицы по первичному ключу;
 - поле первичного ключа является индексом, с помощью которого ускоряется сортировка и поиск записей.
4. Сохранить таблицу (нажать кнопку **«Сохранить»** или **Файл/Сохранить** или знак **«Закрыть таблицу»**).

Для создания таблицы с помощью мастера необходимо:

1. В 1-ом окне мастера выбрать категорию и образец таблицы, затем нужные образцы полей. Допускается выбор полей из нескольких таблиц.
2. Во 2-ом окне мастера выбрать имя для таблицы, а также способ определения ключа. Все сведения, необходимые для создания таблицы, указаны.
3. В 3-ем окне мастера выбирать дальнейшие действия после создания таблицы.
4. Нажать кнопку **Готово**.

Если в дальнейшем потребуется изменить или расширить созданную с помощью мастера таблицу, это можно сделать в режиме конструктора.

ЗАДАНИЕ № 1

Visual FoxPro 9.0

Укажите собственную директорию: **Tools/Options/File Locations/Default Dir...** или задайте в командном окне команду: **Set Default to <имя каталога>**.

Создайте таблицу, содержащую сведения о студентах с именем Stud.dbf (**File-New-Table**), имеющую следующие поля:

| <i>Описание поля</i> | <i>Name</i> | <i>Type</i> | <i>Width</i> |
|--|-------------|-------------|--------------|
| Номер зачётной книжки | Num | Numeric | 10 |
| Фамилия и инициалы | Fio | Character | 35 |
| Дата рождения | DR | Date | |
| Номер группы | N_gr | Character | 10 |
| Год поступления в институт | God | Integer | |
| Вид обучения (бюджет или коммерческое) | Plata | Logical | |
| Количество баллов при поступлении в институт (может быть неизвестно) | Sum_b | Integer | |
| Размер оплаты при поступлении на коммерческой основе | Pay | Currency | |
| Фотография | Photo | General | |
| Адрес | Adr | Memo | |

Обратите внимание, что при интерактивном редактировании данных значение **NULL** вводится нажатием клавиш **CTRL+0**. Для определения отображения пустых значений при работе с окнами Browse, командами DISPLAY и LIST используется команда:

SET NULLDISPLAY TO [текст, отображающий пустое значение]

По умолчанию пустые значения отображаются словом **NULL**.

Заполнить 10-15 записями созданную таблицу.

Просмотреть записи в таблице, используя режим формуляра (**View-Edit**) и табличный режим (**View-Browse**).

Добавить, отметить на удаление и удалить записи таблицы, используя соответствующие пункты меню **Table** для открытой таблицы, находящейся в режиме просмотра.

Обратите внимание на особенность процесса удаления записей в Visual FoxPro, который разбит на 2 этапа:

1. Сначала записи помечаются как удаленные, но физически все еще сохраняются в таблице. Пометку удаления можно снять. Это действие производится командой **RECALL**. Чтобы записи, помеченные как удаленные, не отображались при работе с таблицами, используют специальную глобальную настройку **SET DELETED ON**. **ON** - прячет записи, помеченные как удаленные (учитывает такие записи), а **OFF** - наоборот, отображает такие записи (игнорирует, не учитывает такие записи).

2. Для физического удаления записей необходимо использовать специальную команду **PACK**. Для физического удаления записей в таблице существует еще одна команда: **ZAP**. Данная команда предназначена для физического удаления вообще всех записей таблицы. Причем физическое удаление требует эксклюзивного (единоличного) доступа к таблице. Такой режим открытия таблицы задается либо в команде **USE**, либо в результате установки команды **SET EXCLUSIVE** в **ON**. Записи, удаленные командой **PACK** или **ZAP**, восстановлены быть не могут.

Microsoft Access 2010

Создать одноименную таблицу, содержащую сведения о студентах и подобрав соответствующие типы полей.

Контрольные вопросы

1. Какие действия необходимо выполнить при создании структуры таблицы?
2. Какие основные типы файлов имеются в Visual FoxPro 9.0?
3. Какие типы полей используются в Visual FoxPro 9.0 и Microsoft Access?
4. Что Вы понимаете под полями с NULL значениями?
5. Как осуществляется ввод данных в Мемо – поле в Visual FoxPro 9.0 и Microsoft Access?
6. Когда выбирается режим таблицы, и в каких случаях производится переключение на режим формуляра в Visual FoxPro 9.0?
7. Какие особенности демонстрирует FoxPro при удалении записей, и в чем преимущество существующего механизма удаления?
8. Какая команда выполняет физическое удаление отмеченных записей Visual FoxPro 9.0?

Индексирование таблиц. Теоретические сведения

С целью ускорения поиска и сортировки данных в любой СУБД используются индексы. Индекс является средством, которое обеспечивает быстрый доступ к данным в таблице на основе значений одного или нескольких столбцов. Индекс представляет собой упорядоченный список значений и ссылок на те записи, в которых хранятся эти значения. Чтобы найти нужные записи, СУБД сначала ищет требуемое значение в индексе, а затем по ссылкам быстро отбирает соответствующие записи. Индексы бывают двух типов: простые и составные. Простые индексы представляют собой индексы, созданные по одному столбцу. Индекс, построенный по нескольким столбцам, называется составным. Однако применение индексов приносит не только преимущества, но и недостатки. Главным среди них является тот, что при добавлении и удалении записей или при обновлении

значений в таблице требуется обновлять индекс, что при большом количестве индексов в таблице может замедлять работу. Поэтому индексы обычно рекомендуется создавать только для тех столбцов таблицы, по которым наиболее часто выполняется поиск записей.

Индексирование таблиц в Visual FoxPro 9.0

Для использования алгоритмов поиска, обеспечивающих большую эффективность, а также для связывания нескольких таблиц по общему признаку, выполняется их индексирование по одному полю или комбинации полей (кроме полей типа Memo и General).

В Visual FoxPro можно создавать файлы индексов двух типов:

- Файлы составных индексов .CDX, которые состоят из множества элементов индекса, называемых тегами.
- Файлы индексов .IDX, каждый из которых содержит только один элемент индекса.

Вы также можете создать файл структурного составного индекса, который автоматически открывается вместе с таблицей.

Каждый индекс содержится в отдельном файле, в котором для каждого индекса существует уникальная ссылка, указывающая на местонахождение записи, соответствующей индексу, или является одним из тегов мультииндексного файла.

В Visual FoxPro поддерживаются следующие типы индексов:

Primary (первичный) - используется для создания первичных ключей. Этот индекс может быть только один и не может содержать повторяющихся значений. Поля, входящие в первичный ключ, не должны допускать ввода пустых значений.

Candidate (кандидат) так же, как и первичный индекс, не допускает повторяющихся значений, и обладает всеми качествами первичного ключа и не является им только по той причине, что таблица не может содержать более одного первичного ключа. Любая таблица может иметь более одного индекса этого типа.

Regular (обычный) может содержать повторяющиеся значения, при этом создается индекс, в котором для каждой записи таблицы хранится значение индексного выражения. Используется для установления связи «многие к одному» в той таблице, где находятся «многие».

Unique (уникальный) является обычным индексом, который показывает только одну запись для каждого значения индексного выражения. Уникальный индекс, в отличие от первичного индекса и индекса типа «кандидат», не требует уникальности значений, то есть можно вводить записи с повторяющимися значениями индексного поля. Разница между обычным и уникальным индексом заключается в том, что если две или более записей содержат одинаковое значение индексного выражения, то будет храниться только одно значение и ссылка на первую из записей с

одинаковым значением индексного выражения. Таблица может иметь несколько уникальных индексов.

Если таблица проиндексирована, то такие команды как **DISPLAY**, **LIST**, **BROWSE**, **REPLACE**, **SKIP**, **GO** перемещают указатель записей в соответствии с индексом, а не физическим порядком записей в таблице.

Индексирование и создание индексного файла с расширением .IDX для открытой таблицы выполняется командой

**INDEX ON <врж> TO <имя индексного файла> [FOR <вржL >]
[DESCENDING] [UNIQUE] [ADDITIVE] [COMPACT]**

В качестве <врж> чаще всего используют просто имя поля, по которому упорядочивают таблицы, но индекс может быть и сложным. При этом порядок просмотра записей в таблице определяется полем, имя которого указано в выражении первым.

В команде индексирования опция **DESCENDING** используется для упорядочивания записей по убыванию значений <врж> (по умолчанию – по возрастанию).

Опция **UNIQUE** позволяет игнорировать записи с повторяющимся значением <врж>.

Опция **ADDITIVE** используется для того, чтобы при создании индекса не закрывались уже открытые индексные файлы.

Опция **COMPACT** сохраняет индекслируемые данные в строках фактической, а не фиксированной длины, а повторяющиеся элементы при этом записываются в одном экземпляре, что позволяет сократить размер таблицы в 5-6 раз.

Индексные файлы с расширением .IDX, относящиеся к одной открытой таблице могут быть объединены как теги в один мультииндексный файл с расширением .CDX, который синхронизирует работу сразу нескольких индексов. Мультииндексный файл создается командой

**COPY INDEXES <перечень файлов .IDX > ALL /
[TO < имя неструктурированного файла .CDX >]**

Если опция **TO <>** не используется, то по умолчанию создается структурированный мультииндексный файл, который открывается всякий раз одновременно с таблицей и получает ее имя, но имеет расширение .CDX.

Неструктурированный .CDX - файл не открывается автоматически при открытии таблицы, его нужно открывать специально, если нужен определенный способ упорядочения записей в базе данных.

Для создания нового тега или дополнения файла с расширением .CDX новым тегом используют команду

**INDEX ON <врж> TO TAG < имя тега >
[OF < имя неструктурированного файла .CDX - файла >]
[FOR <вржL >] [DESCENDING] [UNIQUE] [ADDITIVE]**

Индексные файлы с расширением .IDX при активной таблице могут быть открыты командой

SET INDEX TO < список индексных файлов > [ADDITIVE]

Управляющий индекс можно переназначить командой

***SET ORDER TO < имя .IDX – файла > |
< номер .IDX - файла > | [TAG] < имя тега >
[DESCENDING] [ADDITIVE]***

Номер .IDX - файла указывает один из файлов индексов в соответствии с порядком, в котором они перечислены в команде ***USE*** или ***SET INDEX***. Сначала нумеруются открытые файлы .IDX в порядке, в котором они указаны в команде ***USE*** или ***SET INDEX***. Затем нумеруются теги в файле структурного индекса .CDX (если таковой существует) в том же порядке, в каком они создавались. Наконец, нумеруются теги во всех открытых файлах независимых индексов .CDX в том же порядке, в каком они создавались.

Если индексный файл уже создан, его можно открыть одновременно с таблицей командой

***USE < имя таблицы > IN < номер области >
INDEX < список .idx, .cdx – файлов > [AGAIN] [NOUPDATE]***

Подробный формат команд и их описание можно посмотреть в Help.

Индексирование таблиц в Microsoft Access 2010

В Microsoft Access 2010 создать индексы, как и ключи, можно по одному или нескольким полям. Составные индексы позволяют при отборе данных группировать записи, в которых первые поля могут иметь одинаковые значения. Индексировать поля требуется для выполнения частых поисков, сортировок или объединений с полями из других таблиц в запросах. Ключевые поля таблицы индексируются автоматически. Индексировать можно любые поля, кроме MEMO-полей, полей типа Гиперссылка и объектов OLE. Если предполагается, что будет часто выполняться сортировка или поиск одновременно по двум и более полям, можно создать составной индекс. При сортировке таблицы по составному индексу сначала осуществляется сортировка по первому полю, определенному для данного индекса. Если в первом поле содержатся записи с повторяющимися значениями, то сортировка осуществляется по второму полю.

Для создания индекса по одному полю необходимо:

1. В режиме конструктора в панели структуры таблицы (верхняя часть окна) выбрать поле, для которого требуется создать индекс.
2. В панели свойств (нижняя часть окна) для свойства Индексированное поле установить значение «Да (Допускаются совпадения)» или «Да (Совпадения не допускаются)». Убедиться, что в данном поле совпадающих записей нет, можно выбрав значение «Да (Совпадения не допускаются)».

Для создания составного индекса необходимо:

1. В режиме конструктора на панели инструментов нажать кнопку **Индексы**.

2. В первой пустой строке поля **Индекс** ввести имя индекса. Для индекса можно использовать либо имя одного из индексируемых полей, либо другое подходящее имя.

3. В поле **Имя** поля нажать стрелку и выбрать в списке первое поле, для которого требуется создать индекс.

4. В следующей строке поля **Имя** поля указать второе индексируемое поле (для данной строки поле **Индекс** следует оставить пустым). Повторить эти действия для всех полей, которые необходимо включить в индекс. В составном индексе может быть до 10 полей.

В Microsoft Access индексы хранятся в том же файле базы данных, что и таблицы и другие объекты Access.

ЗАДАНИЕ № 2

Visual FoxPro 9.0

Проиндексируйте таблицу Stud.dbf по номеру группы (поле N_gr), создав индексный файл с расширением .idx.

Просмотрите таблицу и выполните все возможные варианты перехода к указанной записи, обратив внимание на то, как изменится перемещение указателя записей.

Проиндексируйте таблицу Stud.dbf по фамилии студента (поле Fio), создав индексный файл с расширением .idx.

Выполните индексирование таблицы Stud.dbf сначала по номеру группы (поле N_gr), а внутри группы по фамилии студента (поле).

Выполните установку индексов с использованием **Table Designer**. Создайте индекс типа «кандидат» по номеру зачетки (поле Num) и обычный индекс по номеру группы (поле N_gr). Обратите внимание, что все теги, созданные в **Table Designer**, автоматически вносятся в структурированный, т.е. одноименный с таблицей мультииндексный файл с расширением .CDX, который используется только для ускорения выполнения таких команд, как **SEEK**, **REPORT** и т.д., а не для визуального упорядочения или выборки данных.

Microsoft Access 2010

Создать аналогичные индексы для таблицы, содержащей сведения о студентах.

Контрольные вопросы

1. Какие типы файлов индексов существуют в Visual FoxPro ?
2. В чём различие между простым индексным и мультииндексным файлами?
3. Какие типы индексов существуют в Visual FoxPro и Microsoft Access?

4. Как осуществляется перемещение указателя записей в проиндексированной таблице?
5. Какой порядок нумерации индексов в Visual FoxPro?
6. Что такое структурированный мультииндексный файл в Visual FoxPro?

Создание баз данных как совокупности взаимосвязанных таблиц. Теоретические сведения

Создание базы данных в Visual FoxPro 9.0

Реляционная база данных – это совокупность взаимосвязанных таблиц, описывающих некоторую предметную область. Под предметной областью понимается часть реального мира, рассматриваемого в данной системе.

При создании базы данных важным понятием является понятие рабочей области. Рабочая область - это область памяти с одним файлом базы данных и несколькими индексными или мультииндексными файлами.

При запуске Visual FoxPro 9.0 по умолчанию активизируется рабочая область 1 (или а).

В указанной рабочей области таблица открывается командой

USE <имя таблицы> IN <номер рабочей области> ALIAS <псевдоним>

Рабочие области и файлы таблиц базы данных могут идентифицироваться псевдонимами. Псевдонимом области по умолчанию является само имя находящегося в ней файла базы данных. В качестве псевдонима можно указать и любое другое слово в команде ***USE***. Область, в которой мы находимся в данный момент, называется активной рабочей областью, и в ней можно работать с находящейся в ней таблицей, выполняя все допустимые команды системы. Одновременно даже в одной команде можно иметь доступ к полям других таблиц базы данных. В этом случае имя поля из неактивной области – составное. Собственно имени поля предшествует имя рабочей области или псевдоним, разделенные знаками «-», или (что более удобно) точкой:

<рабочая область | псевдоним> - <имя поля>

<рабочая область | псевдоним> .<имя поля>

При работе с таблицами, расположенными в нескольких рабочих областях и не связанными между собой, указатели по записям работают независимо друг от друга.

Если открыто несколько таблиц без указания рабочей области, то только последняя будет активной, поскольку будет открыта в рабочей области 1.

Активная рабочая область выбирается командой:

SELECT <номер | имя рабочей области>

| <имя таблицы> | <псевдоним>

Если необходимо открыть несколько таблиц в разных рабочих областях, удобно использовать команду **SELECT 0**, которая автоматически выбирает первую свободную рабочую область.

В Visual FoxPro 9.0 базой данных называется контейнер – таблица с расширением **.dbc**, которая содержит информацию о включенных в нее таблицах, индексах, отношениях между таблицами, представлениях, процедурах и функциях.

База данных позволяет создавать и сохранять отношения, установленные между таблицами, и при необходимости отображать их.

Каждая таблица может принадлежать только одной базе данных.

База данных может быть создана командой

CREATE DATABASE < имя >

или командой **File/New/DataBase/New** меню Visual FoxPro 9.0

Добавить таблицу к базе данных можно любым из трех следующих способов:

1. Выполнить команду меню **DataBase/New** или **DataBase/Add Table**.
2. Щелкнуть на кнопке **New** или **Add Table** панели инструментов **DataBase Designer**.
3. Выбрать команду **New** или **Add Table** в контекстном меню области базы данных.

В Visual FoxPro между таблицами, входящими в базу данных, могут быть установлены связи типа один к одному и один ко многим. Связи типа многие ко многим преобразуются к связям типа один ко многим и многие к одному на более ранних этапах проектирования баз данных. Для установления постоянной связи между двумя таблицами необходимо предварительно проиндексировать каждую из них, причем поле, по которому будет установлена связь, должно быть одного типа и одной размерности. При установлении связи между двумя таблицами одна из них является главной, а другая подчиненной. Главную таблицу называют родительской (это управляющая таблица, то есть Вы манипулируете ею непосредственно), а подчиненную - дочерней (или таблицей-потомком, которой управляют через связь с родителем).

Для установления связи 1:1 типы индексов в каждой из связываемых таблиц должны быть или типа **Candidate** или типа **Primary**. Для установления связи 1:N тип индекса родительской таблицы (соответствующей односвязной сущности) должен быть или **Primary** или **Candidate**, а дочерней (соответствующей N - связной сущности) или **Regular** или **Unique**.

Для создания отношений между таблицами, включенными в базу данных, необходимо подвести мышь к индексному полю родительской таблицы, нажать на это поле и, не отпуская перенести на соответствующее индексное поле дочерней таблицы, по которому устанавливается связь. Таким образом устанавливаются необходимые связи между всеми таблицами, входящими в базу данных.

Необходимо отметить, что отношения, созданные в *Конструкторе баз данных* сохраняются и после окончания работы с системой, требуют индексации всех таблиц, и не управляют синхронным перемещением курсора, а могут быть использованы для создания представлений и запросов.

Для установления временных связей между таблицами, то есть связей существующих во время одного сеанса работы с FoxPro, необходимо также проиндексировать связываемые таблицы по общим полям, затем выполнить последовательность действий – *Window/Data Session* и в окне *Data Session* открыть нужные таблицы. Для установления связи 1:1 подвести курсор к родительской таблице, нажать кнопку *Relations* и подвести курсор к дочерней таблице, после выбора, которой в правой части окна будет показана установленная связь. Для установления временной связи 1:N необходимо установить связь 1:1, затем подвести курсор к родительской таблице и нажать на кнопку «*1 To Many*», после чего в правой части окна будет показана связь типа 1:N между соответствующими таблицами. При просмотре связанных таблиц переход по записям осуществляется синхронно, с учётом установленных связей.

Создание базы данных в Microsoft Access 2010

Механизм описания логических связей между таблицами в Access реализован в виде объекта, называемого *Схемой данных*. Перейти к ее созданию можно из панели инструментов *Работа с базами данных/Схема данных*, доступной из главного окна.

Для добавления таблицы в схему базы данных необходимо:

1. Щелкнуть правой кнопкой мыши на свободном пространстве схемы данных и из контекстного меню выбрать команду *Добавить таблицу*.
2. В диалоговом окне *Добавление таблицы* раскрыть вкладку *Таблицы* и выбрать одну или несколько таблиц из списка таблиц, и нажать кнопку *Добавить*.

Для установления связей между таблицами необходимо:

1. Перетащить поле (как правило, поле первичного ключа) из одной таблицы на общее поле (поле внешнего ключа) в другой таблице. Чтобы перетащить сразу несколько полей, нажмите клавишу *CTRL* и, удерживая ее, выберите каждое поле.
2. В открывшемся диалоговом окне *Изменение связей* при необходимости задайте условия обеспечения целостности данных и объединения и нажмите кнопку *ОК*.

Access проведет линию связи между двумя таблицами. Если был установлен флажок *Обеспечение целостности данных*, линия будет толще на каждом конце. Кроме того, если был установлен флажок *Обеспечение целостности данных*, над широкой частью на одном конце линии связи

будет отображено число 1, а над широкой частью на другом конце линии — символ бесконечности (∞).

Для создания отношения «один-к-одному» оба общих поля (как правило, поля первичного ключа и внешнего ключа) должны иметь уникальный индекс. Это означает, что свойство **Индексированно** этих полей должны иметь значения **Да (Совпадения не допускаются)**. Если оба поля имеют уникальный индекс, Access создаст отношение «один-к-одному».

Для создания отношения «один ко многим» поле на одной стороне отношения (как правило, поле первичного ключа) должно иметь уникальный индекс. Это означает, что свойство **Индексировано** этого поля должны иметь значения **Да (Совпадения не допускаются)**. Поле на стороне «многие» не должно иметь уникального индекса. У этого поля может быть индекс, но он должен допускать совпадения. Это означает, что свойство **Индексировано** этого поля должно иметь значение **Нет** либо **Да (Допускаются совпадения)**. Если одно поле имеет уникальный индекс, а другое — нет, Access создаст отношение «один ко многим». Для изменения связи между таблицами следует выбрать ее на вкладке **«Схема данных»**, а затем изменить эту связь.

ЗАДАНИЕ № 3

Visual FoxPro 9.0

Создать базу данных Base1.DBC, хранящую информацию о студентах, группах, изучаемых дисциплинах и паспортные данные студентов. Для этого добавьте созданную ранее таблицу Stud.dbf и создайте следующие новые таблицы, самостоятельно определив тип и размерность полей:

Паспортные данные (номер зачетки, серия паспорта, номер паспорта, кем выдан, дата выдачи);

Группа (номер группы, специальность);

Дисциплина (код дисциплины, название дисциплины);

Учебный_план (номер группы, код дисциплины, № семестра, кол-во часов, вид аттестации);

Аттестация (номер зачетки, код предмета, дата, оценка).

Установите следующие связи между таблицами:

1. Группа - Студент – 1:N;
2. Студент - Паспортные данные – 1:1;
3. Группа - Учебный_план 1:N;
4. Дисциплина - Учебный_план 1:N;
5. Студент - Аттестация - 1: N.

Используя **Data Session**, установить следующие временные связи между таблицами, определив тип связи по множественности:

1. Группа → Студент → Паспортные данные → Аттестация
2. Группа → Учебный_план → Дисциплина
3. Дисциплина → Учебный_план → Группа → Студент → Паспортные данные → Аттестация

Просмотреть через Browse, как установленные связи отображаются при движении по записям в таблицах.

Microsoft Access 2010

Создать аналогичную базу данных средствами Microsoft Access. Реализовать различные варианты просмотра данных, задавая имя подтаблицы в окне свойств, в режиме ***Конструктор таблицы***. Перейти к нему можно из панели инструментов ***Работа с базами данных/Страница свойств*** главного окна.

Контрольные вопросы

1. Что такое рабочая область в Visual FoxPro?
2. Как создать базу данных, установив постоянные связи между таблицами в Visual FoxPro и Microsoft Access?
3. Какие типы связей реализуются в Visual FoxPro и Microsoft Access?
4. В чём отличие связей один к одному и один ко многим?
5. Как реализовать связи многие ко многим?
6. Как организовать временную связь между файлами базы данных в Visual FoxPro?

ЛАБОРАТОРНАЯ РАБОТА № 2

КОМАНДЫ ПЕРЕМЕЩЕНИЯ В ТАБЛИЦЕ, ПОИСК ЗАПИСЕЙ В НЕУПОРЯДОЧЕННОЙ ТАБЛИЦЕ И ОРГАНИЗАЦИЯ ЗАПРОСОВ

Команды перемещения в таблице, поиск записей в неупорядоченной таблице. Теоретические сведения

Visual FoxPro 9.0

Непосредственно после открытия таблицы (команда **USE**) указатель устанавливается на первую запись, которая становится текущей и подлежит обработке. Для работы с записью или группой записей используют команды перемещения в таблице или установки указателя.

Команда **GO TOP** устанавливает указатель на первую запись таблицы (то же выполняет безоперандная команда **LOCATE**).

Команда **GO BOTTOM** перемещает указатель на последнюю запись таблицы.

Команда **GO N** устанавливает указатель на запись таблицы с номером N.

Команда **SKIP <N>** перемещает указатель на запись таблицы, отстоящую от текущей на N записей. При этом N может быть как положительным, так и отрицательным числом.

Для перехода к записи необходимо набрать соответствующую команду в командном окне или необходимо выполнить следующую последовательность действий для открытой таблицы, находящейся в режиме просмотра – **Table/Go to Record**. При этом возможны следующие варианты перехода: **Top** - к первой записи, **Bottom** - к последней, **Next** - к следующей от текущей (текущей записью считается запись, на которой стоит курсор), **Previous** - к предыдущей, **Record #** - к записи с указанным номером, **Locate** – последовательный поиск первой записи, удовлетворяющей заданному условию. В диалоговом окне для описания условия поиска указывается **Scope**-область поиска (**All** - поиск по всем записям, **Next+число** - поиск от текущей записи указанное число записей, **Record+число** - поиск от начала таблицы указанное число записей, **Rest** - поиск от текущей записи до конца таблицы), условие **For** (поиск записей, для которых выполняется введённое условие) и/или условие **While** (поиск записей до тех пор, пока выполняется введённое условие), определяемых с помощью диалогового окна конструктора выражений.

Совместно с **LOCATE**, для продолжения поиска записей, удовлетворяющих условию, используется команда – **Continue**.

При неудачном поиске указатель перемещается в конец таблицы и функция **EOF()** возвращает логическое значение **.T.**, а **FOUND()** – значение **.F..**

Если поиск записи выполняется по значению символьного поля, то при предварительной установке **SET EXACT ON** поиск первой подходящей

записи выполняется по всей искомой строке, а если используется установка **SET EXACT OFF**, то поиск может производиться по начальным символам строки.

Для выделения группы записей, удовлетворяющей требуемым условиям, в Visual FoxPro 9.0 широко используются фильтры.

Команда **SET FILTER TO <вржL>** выделяет в таблице группу записей, для которых результатом вычисления заданного логического выражения является **.T.** В свою очередь логическое выражение записывается в формате: **<выражение любого типа> = <значение такого же типа>**. После установки фильтра остальные записи становятся недоступными. Для выполнения команды **SET FILTER TO <вржL>** с использованием фильтра следует предварительно выполнить любую команду, перемещающую указатель в таблице.

Функция **FILTER()** возвращает установленный фильтр для текущей рабочей области.

Для снятия фильтра выполняется безоперандная команда **SET FILTER TO**.

Установить фильтр можно с помощью команды **SET FILTER TO <вржL>**, заданной в командном окне, или при помощи команды системного меню **Table/Properties/Work Area Properties/Data Filter**.

Наиболее часто поиск данных в отдельных и взаимосвязанных таблицах осуществляется с помощью языка SQL, используемого для описания запросов, который будет рассмотрен позднее.

Microsoft Access 2010

Текущая запись в режиме Таблицы в Microsoft Access выделяется цветом. Для перемещения на нужную запись, по записям, видимым на экране, надо просто щелкнуть линию на любом поле этой записи. Кроме того, в нижней части окна расположены кнопки навигации, позволяющие перемещать указатель текущей записи по таблице (на первую запись, на предыдущую запись, на следующую запись, на последнюю запись). Там же находятся поле номера текущей записи, кнопка создания новой записи и указатель общего количества записей в таблице, индикатор фильтра, поле поиска, которое можно использовать для быстрого поиска записи с совпадающим значением. При вводе текста в поле **Поиск** по мере ввода каждого символа первое совпадающее значение выделяется в режиме реального времени.

Microsoft Access позволяет производить поиск данных в активных таблицах по следующим условиям:

1. По всем полям каждой записи или отдельному полю всей записи.
2. Вперед или назад.
3. По типу сравнения данных.

Для поиска данных по таблице можно использовать диалог **«Поиск и замена»**. Данное диалоговое окно можно использовать, только если в текущий момент в таблице или форме отображаются данные. Это

распространяется даже на случай, если видимых записей нет из-за применения фильтра. Вызвать данный диалог можно используя пункты меню *Главная/Найти*, или сочетанием клавиш **<Ctrl> + <F>**. Откроется диалоговое окно *«Поиск и замена»*. Курсор будет находиться на вкладке *«Поиск»*. В поле *«Образец»* введите значение для поиска. При необходимости можно воспользоваться списком *«Поиск в»*, чтобы сменить поле для поиска или выполнить поиск по всей базовой таблице.

Список *«Совпадение»* представляет собой оператор сравнения (такой как «равно» или «содержит»). При необходимости в списке *«Совпадение»* выберите пункт *«с любой частью поля»*. Это обеспечивает более широкий поиск.

Для отображения записей, соответствующих определенным условиям и операторам сравнения Microsoft Access использует фильтрацию.

Фильтрация — удобный способ отображения нужных данных. Фильтры позволяют просмотреть только отдельные записи в таблице, форме, отчете, или запросе либо напечатать некоторые записи из таблицы, отчета, или запроса. С помощью фильтра можно ограничить объем отображаемых данных, не изменяя макет базовых объектов. В отличие от поиска, фильтрация только ограничивает отображаемые записи.

В Microsoft Access существуют следующие типы фильтров:

- **Обычные фильтры**: используются для фильтрации по значению или диапазону значений. Они используются для всех типов полей за исключением полей объектов OLE и полей, в которых отображаются вычисленные значения. Обычные фильтры встроены в каждое представление Access 2010. Доступность команд фильтра зависит от типа и значений поля. Чтобы отфильтровать таблицу по нескольким столбцам, необходимо либо выбрать и отфильтровать каждый из них по отдельности, либо воспользоваться расширенным фильтром.
- **Фильтрация по выделенному**: позволяет отобразить все строки в таблице, содержащие значение, которое совпадает с выделенным значением в строке. Используется в режиме таблицы. Для просмотра всех строк таблицы, содержащих значение, которое совпадает с выделенным нужным значением, надо щелкнуть нужную ячейку правой кнопкой мыши. В раскрывающемся списке будут показаны доступные параметры фильтрации, которые зависят от типа данных выделенного значения. Чтобы применить фильтр по частичному выделению, выделите только нужные знаки.
- **Фильтр по форме**: используется, если требуется отфильтровать несколько полей в форме или таблице либо найти конкретную запись. В Access создается пустая форма или таблица, аналогичная исходной, после чего можно заполнить в ней любые

поля. Будут найдены записи, в которых содержатся указанные значения.

Иногда может возникнуть необходимость в определении нескольких различных наборов условий. Чтобы получить все записи, соответствующие любому из нескольких наборов условий, введите первый набор условий. Затем откройте вкладку **Или** и введите следующий набор условий. Вкладка **Найти** и каждая вкладка **Или** задают отдельный набор значений для фильтра. Каждый раз, когда на вкладку **Или** добавляется условие, создается другая вкладка **Или**, что позволяет указать несколько альтернативных условий отбора.

— **Расширенный фильтр**: позволяет задать пользовательские условия фильтра.

Для применения расширенных фильтров необходимо создавать выражения. Выражения аналогичны формулам в Excel и условиям, которые задаются при создании запроса.

Необходимо отметить, что некоторые символы, например *, % и ?, в текстовом поле фильтра считаются специальными знаками. Так, звездочка (*) представляет строку знаков, поэтому строка «a*» соответствует любой строке, начинающейся с буквы a, а не только строке a*. Если не требуется, чтобы знак считался специальным, необходимо заключить его в квадратные скобки ([]), например «a[*]». В базах данных, использующих стандарт ANSI-89, к специальным относятся знаки *, ?, [,], !, - и #. В базах данных стандарта ANSI-92 специальными считаются знаки %, _, [,], ^ и - В Access можно использовать любой из этих стандартов, но не оба стандарта одновременно (например, нельзя использовать в выражении сочетание ?a*).

Для задания условий фильтрации, которые соответствуют условиям запроса используется команда **Загрузить из запроса**.

Команда **Сохранить как запрос** позволяет сохранить параметры фильтра в качестве нового запроса.

ЗАДАНИЕ №1

Visual FoxPro 9.0

1. Выполнить все возможные варианты перехода к указанной записи.
2. Осуществить последовательный поиск записей:
 - 2.1. Найти студентов, год поступления которых в институт равен заданному и учатся на бюджете.
 - 2.2. Найти студентов нескольких заданных групп, которые родились в указанном месяце.

- 2.3. Найти студентов, которые поступили на бюджет и у которых количество баллов при поступлении неизвестно.
- 2.4. Найти студентов, у которых количество баллов при поступлении больше заданного значения и учатся на коммерческой основе.
- 2.5. Осуществить поиск записей по начальным буквам фамилии, и по полному совпадению, используя установки **SET EXACT OFF** и **SET EXACT ON**.

Для продолжения поиска в командном окне набрать команду *Continue*, для просмотра *Browse*.

3. Выделить группы записей, по условиям, перечисленным в пунктах 2.1, 2.2, 2.3, 2.4, установив соответствующие фильтры.

Microsoft Access 2010

1. Выполнить все возможные варианты перехода к указанной записи.
2. Осуществить поиск данных по таблице с использованием диалога «Поиск и замена».
3. Выделить группы записей, по условиям, перечисленным в пунктах 2.1–2.4 задания для Visual FoxPro, установив соответствующие фильтры.

Контрольные вопросы

1. Что понимается под последовательным поиском данных?
2. Как осуществляется переход к следующей искомой записи при стандартном поиске в Visual FoxPro?
3. Как осуществляется поиск в Microsoft Access?
4. Какая функция возвращает результат выполнения поиска в Visual FoxPro?
5. Как влияют на результат поиска установки SET EXACT OFF и SET EXACT ON в Visual FoxPro?
6. В каких случаях необходимо использовать фильтры?
7. Какая команда используется для снятия фильтра в Visual FoxPro?
8. Какие типы фильтров существуют в Microsoft Access?

Организация запросов. Теоретические сведения

Visual FoxPro 9.0

Наиболее часто поиск данных в отдельных и взаимосвязанных таблицах осуществляется с помощью языка SQL (Structured Query Language), используемого для описания запросов.

Основные команды языка SQL – *Create, Insert, Delete, Update, Select*. Команда *Select* используется при организации запросов к базе данных.

В Visual FoxPro для организации запросов имеются удобные инструментальные средства.

Чтобы создать достаточно сложные запросы, следует использовать *Query Designer*. Для этого выберите *File/New/Query* или выполните команду

CREATE QUERY [<имя запроса>] [NOWAIT]

SQL-выражение, создаваемое при работе в конструкторе запросов, сохраняется в файле с расширением .QPR – его можно запускать и редактировать как обычный .PRG – файл, выполнив команду

MODIFY COMMAND <имя файла>.qpr или
MODIFY FILE <имя файла>.qpr,

а также можно сохранить и в любой момент времени выполнить с помощью команды **DO**.

Окно конструктора запросов состоит из области представления таблиц и отношений между ними, занимающей верхнюю часть основного окна, и области для записи условия и выводимых в запросе полей, занимающей нижнюю часть окна и включающую шесть вкладок:

- **Fields**-выводимые в запросе поля;
- **Join** – условие соединения таблиц;
- **Filter** – условие выборки;
- **Order By** – упорядочение;
- **Group By** – группировка;
- **Miscellaneous** – дополнительные опции запроса.

Чтобы добавить таблицу в область представления таблиц, щелкните правой клавишей мыши в любом месте верхней области, и в появившемся меню выберите команду **Add Table**. В запрос могут входить как таблицы в составе базы данных, так и свободные таблицы. При открытой базе данных также можно выбрать и включить в запрос представления (**View**). При формировании запроса Вы можете устанавливать отношения между таблицами независимо от наличия индекса. Поскольку Вы просто указываете критерии объединения, Visual FoxPro не требует наличия соответствующих индексов. Очевидно, связывание двух таблиц по полям, не имеющим ничего общего, даст бессмысленный результат.

Вкладка **Filter** служит для построения выражения, используемого в предложении **Where**. Для построения сложных выражений можно щёлкнуть кнопку **Logical**, которая вставит оператор **OR**. Кнопки **Add And** нет. Так как **AND** используется по умолчанию для соединения частей выражения. Причём первые несколько выражений, определяющих условия объединения, редактируются иначе, чем условия дополнительной фильтрации данных, попадающих в результат выполнения запроса. Редактировать их напрямую нельзя. Предложение **HAVING** добавляется щелчком кнопки **Having** на вкладке **Group By**. При формировании условия можно использовать следующие критерии:

- **Like** (похоже) – содержимое поля должно быть похоже на образец;
- **Exactly Like** – содержимое поля должно точно совпадать с образцом;
- **More Than** – значение слева больше чем значение справа;
- **Less Than** – значение слева меньше чем значение справа;

- **Between** (между) – вхождение в диапазон;
- **In** – вхождение в множество;
- **Equal** – значение слева равно значению справа;
- **Is Null** (не определено) – проверяет неопределенное (не нулевое) значение. Пустые символьные строки, нули, пустые даты не являются неопределенными, неопределенное значение – это значение, которое не задавалось.

Вкладка **Fields** позволяет определить, какие поля из таблиц будут включены в результат выполнения запроса. В результат можно включить любые поля из исходных таблиц, а также выражения с использованием функций **SUM()**, **AVG()**, **COUNT()**, **MIN()**, **MAX()**.

Вкладка **Order By** позволяет упорядочить выборку по одному или нескольким полям/выражениям, включенным в результат запроса. Нельзя выполнить сортировку по полю, не включенному в список выходных полей.

Вкладка **Group By** в отличие от вкладки **Order By** кроме списка полей предлагает набор агрегатных функций – **SUM()**, **AVG()** и т.д. На этой вкладке расположена кнопка, добавляющая в выражение SQL предложение **HAVING**, которое используется для назначения условий фильтрации, которым группа должна удовлетворять, чтобы попасть в результат выполнения запроса.

При работе с конструктором запросов, в основном меню Visual FoxPro появляется пункт меню **Query**, включающей команды- **Add Table** (добавляет таблицу в запрос), **Remove Table** (удаляет таблицу из запроса), **Remove Join Condition** (удаляет условие объединения таблиц), **Selection Criteria** (открывает вкладку **Selection Criteria**), **Output Fields** (открывает вкладку **Fields**), **Order By** (открывает вкладку **Order By**), **Group By** (открывает вкладку **Group By**), **Query Destination** (выбор направления вывода результатов запроса), **View SQL** (показывает команду **Select**, соответствующую сконструированному инструментальными средствами запросу), **Comments** (для связи комментария с запросом), **Run Query** (выполнить запрос).

Выбор команды **Output Destination** из меню выводит диалоговое окно, содержащее варианты различных направлений вывода результатов запроса. При выборе **Browse** (по умолчанию) запрос выводится на экран в виде таблицы, **Graph** – в графическом виде, **Table** – выводится в файл DBF, **Screen** – вывод на принтер или в текстовый файл, **Report** – вывод в отчет, **Label** – вывод в этикетку, **Cursor** – вывод в курсор, то есть временный файл, который можно просматривать, по записям которого можно двигаться в любом направлении, осуществляя новые выборки.

Microsoft Access 2010

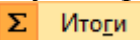

Для получения необходимых для пользователя данных из таблиц в Access 2010 включён самостоятельный объект – **Запросы**. Запросы создаются с помощью Мастера запросов, Конструктора запросов и языка запросов SQL.

Для создания запросов к базе данных с помощью конструктора запросов на вкладке «*Создание*» выбрать режим «*Конструктор запросов*».

Далее для формирования простого запроса в режиме Конструктор необходимо выбрать таблицы, которые будут участвовать в запросе, и заполнить бланк запроса, служащий для определения параметров запроса, для этого в бланке запроса необходимо указать наименование поля таблицы, из которой это поле выбирают. На бланке запроса в строке под наименованием «*Вывод на экран*» установлен знак , в отдельных случаях можно снять эту пометку, тогда данные из этого поля будут обрабатываться в запросе, но на экране мы их не увидим. Если необходимо установите параметры объединения таблиц, для чего выделите линию связи между таблицами, щёлкните правой кнопкой мыши по этой линии. В контекстно-всплывающем меню выберите строку с командой *Параметры объединения*. Запрос можно выполнить или сохранить для дальнейшего использования.

Для создания параметрических запросов, т.е. запросов, в которых отбор значений данных из таблиц осуществляется по заданному параметру необходимо задать параметры отбора.

Следует отметить, что параметры для отбора данных могут быть заданы в виде текста или числа, с помощью диалогового окна или выражения. При вводе жёсткого параметра в виде текста, тогда текстовый параметр заключается в двойные кавычки («...»). При вводе жёсткого параметра в виде числа, двойные кавычки не используют. При использовании диалогового окна, признаком его открытия являются открытая и закрытая квадратные скобки ([...]), параметр, вводимый в диалоговое окно может быть и текстом и числом. При построении логического выражения в качестве параметра, следует помнить, что сравниваются только данные, принадлежащие одному столбцу бланка запроса.

Для добавления в бланк запроса строки с наименованием «*Групповая операция*» можно щёлкнуть правой кнопкой мыши по бланку запроса, а затем выбрать строку  или на ленте активизировать элемент  - итоги (на вкладке *Работа с запросами*).

ЗАДАНИЕ №2

Формирование запроса по неполному значению поля

Пользователь, обращаясь к базе данных, может предполагать лишь приблизительное написание текстовых значений, например, фамилии, или предполагать диапазон численных данных, которые его интересуют. В этом случае создают запрос на выборку по неполному значению поля, при этом целесообразно составлять поисковое предписание, использующее стандартные функции «*Like*» – Подобно и «*Between*» – Между.

1. Вывести списки студентов с указанием № группы, ФИО студента, даты рождения, которые родились в указанный в годах промежуток времени и обучаются в указанных группах. Список упорядочить по № группы и дате рождения.

2. Вывести списки студентов с указанием № группы, ФИО студента, чьи фамилии начинаются на заданную букву и обучаются в указанных группах.

3. Сформулировать и реализовать свой запрос.

Выполнить запросы средствами Visual FoxPro и Access 2010.

ЗАДАНИЕ № 3

Формирование запроса с использованием параметров

1. Вывести список студентов, которые родились в заданном, через параметр месяце. Список упорядочить по номеру группы.

2. Вывести список студентов, которые родились в заданном году и чьи фамилии начинаются с заданной буквы. Год и букву задать через параметр.

3. Сформулировать и реализовать свой запрос.

При вводе параметров в Visual FoxPro использовать либо форму для ввода параметров, либо функцию *INPUTBOX(.)*.

Выполнить запросы средствами Visual FoxPro и Access 2010.

ЗАДАНИЕ № 4

Запросы с использованием NULL значений

1. Вывести списки студентов с указанием № группы, ФИО студента, у которых неизвестно количество баллов при поступлении и обучаются в указанных группах. Список упорядочить по № группы, ФИО.

2. Вывести списки студентов, у которых неизвестно количество баллов при поступлении или неизвестна дата рождения и обучаются на коммерческой основе.

3. Сформулировать и реализовать свой запрос.

Выполнить запросы средствами Visual FoxPro и Access 2010.

ЗАДАНИЕ № 5

Многотабличные запросы

1. Вывести списки студентов, указав ФИО и специальность, по которой они обучаются. Список упорядочить по специальностям, а затем в алфавитной последовательности.

2. Вывести списки студентов, у которых отсутствуют паспортные данные. При создании запроса использовать оператор *Not IN*.

3. Вывести список студентов, обучающихся по заданной через параметр специальности, указав номер зачетной книжки, ФИО и № группы. Список упорядочить по № группы и ФИО.

4. Вывести результаты аттестации по заданной через параметр дисциплине. Список упорядочить по оценке.

5. Сформулировать и реализовать свой запрос.

Выполнить запросы средствами Visual FoxPro и Access 2010.

ЗАДАНИЕ № 6

Запросы с использованием агрегатных функций и группировкой

1. Подсчитать количество студентов, обучающихся по каждой специальности. Посмотреть, как изменится результат выполнения запроса в зависимости от параметров объединения таблиц.

2. Подсчитать среднюю оценку по каждой дисциплине для каждой группы.

3. Вывести списки дисциплин, средняя оценка по которым больше указанной величины.

4. Вывести списки групп, для которых средний балл студентов при поступлении меньше заданного через параметр значения.

5. Сформулировать и реализовать свой запрос.

Выполнить запросы средствами Visual FoxPro и Access 2010.

Контрольные вопросы

1. Как отображаются результаты запроса?

2. Какие средства предложены в Visual FoxPro и Access 2010 для создания запросов?

3. Какие основные типы запросов создают в базе данных Access 2010?

4. Что подразумевают под словом «Параметры» при создании запроса?

5. Можно ли задавать несколько параметров в одном запросе для различных полей?

6. Что обозначают открытая и закрытая квадратные скобки в выражении?

7. Как проектируются запросы к нескольким таблицам одновременно?

8. Когда в запросах рекомендуется группировать данные?

9. Какие варианты вывода результатов запросов Вы знаете?

10. Поясните смысл каждого раздела команды Select, обозначенного цифрами: **SELECT** 1 **FROM** 2 **WHERE** 3 **GROUP BY** 4 **HAVING** 5 **ORDER BY** 6

11. Возможно ли использование функций в запросах?

ЛАБОРАТОРНАЯ РАБОТА № 3

СОЗДАНИЕ ОТЧЕТОВ

Отчет – это форматированное представление данных, которое выводится на экран, в печать или файл.

Теоретические сведения

Visual FoxPro 9.0

В Visual FoxPro 9.0 для создания отчетов можно использовать мастера отчетов (*Report Wizard*) и конструктор отчетов (*Report Designer*).

Для создания отчета с использованием мастера отчета необходимо выполнить команду *Tools/Wizard/Report*. Для построения отчета по одной таблице используется *Report Wizard*. Если же в отчете будут поля из связанных таблиц, выберите *One-To-Many Report Wizard*.

Мастер отчетов строит отчет по шагам, каждый из которых описан в мастере, поэтому работа с мастером проста и здесь подробно не описывается.

Созданный отчет сохраняется в файлах описания с расширением *.frx* и *.frt*, которые можно использовать в программе для вывода полученного отчета на принтер, экран (для просмотра) или в текстовый файл. Для этого необходимо выполнить команду

***REPORT FORM <имя отчета> TO PRINTER [PROMPT]
/ PREVIEW / TO FILE <имя файла> [ASCII]***

Опция PROMPT выводит на экран диалоговое окно Print, в котором можно выполнить установки, определяющие работу печатающего устройства.

Если используется вывод отчета в файл (*TO FILE*), то по умолчанию создается файл с расширением *.txt*.

При выводе исключительно текстовой информации применяется опция ASCII.

Запуск конструктора отчетов можно осуществить одним из следующих способов:

- Выполните команду меню *File/New/Report/ New*
- В командном окне ведите команду ***CREATE REPORT <имя отчета>***.

При работе с отчетом в основное меню добавляется пункт *Report* для выполнения основных операций при конструировании:

- *Title/Summary*-используется для формирования первой и последней полос отчёта.
- *Data Grouping* - позволяет визуально группировать данные для сортированных и индексированных данных. Например, можно распечатать список студентов для группы имен, начинающихся с буквы 'А', затем с 'В' и так далее, подобно справочнику.

- **Variables** (переменные) - позволяет включить в отчет данные, не представленные в таблице. Например, чтобы создать общее количество студентов в списке нужно создать переменную, где будет храниться это число.
- **Default Font** - определяет шрифт, которым будет выводиться каждый вновь создаваемый объект отчёта.
- **Private Data Session** - означает, что Вы хотите иметь независимый указатель записи, положение которого при печати отчета никак не отражается на положении указателя в той же самой таблице для остальных элементов программы. Это позволит, например, просматривать данные по студентам во время вывода отчета на печать.
- **Quick Report** - используется для быстрого создания несложных отчетов.
- **Run Report** – проверка функциональности отчёта не покидая окно конструктора.

В окне конструктора отчетов имеются следующие типы полос:

- **Title** (Титул)

В этой полосе размещается информация, появляющаяся перед основным отчетом и называемая титульной. Её можно вынести на отдельную страницу, для этого нужно выбрать команду Title/Summary в меню Report и установит флажок New Page.

- **Page Header** (Верхний колонтитул)

Данные, помещенные в полосу, печатаются в начале каждой страницы. Например, в ней размещаются имена полей, которые выводятся как заголовки содержимого полосы Detail, и пользователь легко понимает, что представляют собой данные в теле отчета.

- **Detail** (Детали)

Эта полоса содержит данные полей из таблицы или результат вычисления над ними. Вы можете раздвинуть её до получения нужной высоты и поместить туда несколько строк данных. При необходимости можно добавить в полосу прямоугольники или линии для повышения читаемости отчёта или поместить туда дополнительный текст комментариев.

- **Page Footer** (Нижний колонтитул).

Используется для печати внизу каждой страницы отчета некоторых итогов. Для печати общего итога отчёта можно использовать полосу **Summary**, которая открывается из меню при выборе соответствующего пункта.

- **Group Header** (Группа сверху)

В этой полосе печатается информация, используемая при группировке и позволяющая идентифицировать информацию, содержащуюся на каждом уровне группировки.

- **Group Footer** (Группа снизу)

В полосе размещается итоговая информация по группе.

— **Summary (Итогу).**

При работе с отчётом, щёлкнув в любом месте отчета правой кнопкой мыши, не занятом объектами, появится всплывающее меню, содержащее следующие команды: **Paste, Preview, Print, Data Environment, Grouping, Help**. Для включения таблиц в проектируемый отчет используется, например, команда **Data Environment**, описывающая окружение данных отчёта.

Для создания вычисляемых полей нужно щёлкнуть по полю отчёта правой кнопкой мыши, в окне конструктора выражения выбрать нужную функцию и построить требуемое арифметическое выражение для вычисляемого поля.

Этикетка – специальный вид отчёта, используемый, как правило, для печати адресов и сведений об адресатах, которые затем наклеиваются на конверты.

Для создания этикеток применяется мастер этикеток Label Wizard или конструктор этикеток Label Designer.

В результате создаются файлы описания этикеток, которые имеют расширение .ltx или .lbt.

Для работы с этикетками используются те же команды, что и для отчетов, но слово REPORT заменяется словом LABEL.

Microsoft Access 2010

В приложении Access отчет состоит из разделов:

1. Раздел заголовка может отображаться в верхней части отчета; раздел заголовка страницы отображается в верхней части каждой страницы отчета. В заголовках отчетов, как правило, отображаются названия и изображения, например, логотип компании. В заголовках страниц обычно отображаются заголовки столбцов.
2. Для группировки данных в отчете используется Заголовок группировки. В этом разделе содержатся поля, по которым можно группировать данные. В этом разделе представлены поля, по которым группируются данные.
3. В разделе область данных содержится основной текст отчета — все важные данные. Этот раздел обязателен во всех отчетах.
4. Разделы примечаний могут отображаться в нескольких местах. Например, можно создать примечания групп, в которых будут отображаться итоги, подсчеты или средние значения для группы данных. Можно также создать нижние колонтитулы, например, для отображения номеров страниц отчета.

В Microsoft Access 2010 можно создать простой отчет на основе данных текущего запроса или таблицы. Для этого:

1. В области навигации выберите таблицу или запрос, который требуется использовать в качестве источника данных для отчета.
2. На вкладке Создание в группе Отчеты нажмите кнопку Отчет. Access создаст отчет, включающий в себя все поля источника данных. В отчете автоматически будет использоваться табличный

макет, и если источник данных содержит достаточно полей, Access создаст отчет с альбомной ориентацией. Отчет открывается в режиме макета, в котором можно вносить изменения.

Созданный таким образом отчет можно доработать, добавив дополнительные компоненты, например группы или итоги.

Создать простой настраиваемый отчет можно также с помощью мастера отчетов. Для этого:

1. На вкладке **Создание** в группе **Отчеты** нажмите кнопку **Мастер отчетов**.
2. Выполните предлагаемые мастером действия.
3. Если результат вас не устроит, удалите отчет и начните все заново, выбирая другие параметры.

Для создания отчета с нуля необходимо:

1. На вкладке **Создание** в группе **Отчеты** нажать кнопку **Пустой отчет**. Access создаст пустой отчет и отобразит область задач **Список полей**.
2. В области задач **Список полей** открыть таблицу, которая будет служить источником данных, а затем перетащить поля в отчет. Access автоматически применит табличный макет.
3. Если вы хотите использовать другой макет, выделите поля в отчете, щелкнув элемент выделения макета (крестик в его левом верхнем углу) и на вкладке **Упорядочение** в группе **Таблица** нажмите кнопку **В столбик**.

Наиболее часто отчеты создаются с помощью конструктора отчета. Такие отчеты создаются с «нуля» или дорабатываются отчеты, созданные **Мастером отчетов**. Для создания нового пустого отчета в режиме конструктора необходимо на вкладке **Создание** в группе **Отчеты** нажать кнопку **Конструктор отчетов**.

Для создания средствами конструктора отчета с параметрами используется не таблица, а ранее созданный параметрический отчет. При запуске такого отчета на просмотр сначала появляется диалоговое окно для ввода изменяемого значения (параметра), затем сам отчет. Вид отчета меняется в зависимости от введенного значения параметра.

Для создания средствами конструктора отчета с группировкой и итогами добавляется Полоса группировки.

Для группировки данных в отчете необходимо:

1. Открыть отчет в режиме макета или конструктор, а затем на вкладке **Конструктор** в группе **Группировка и итоги** нажать кнопку **Группировка**. Под отчетом появится область **Группировка, сортировка и итоги**.

2. В этой области нажать кнопку **Добавить группировку**, а затем выбрать поле, по которому требуется сгруппировать данные. Access сгруппирует данные в соответствии с вашим выбором.
3. Если вы хотите выполнить сортировку данных, нажмите кнопку **Добавить сортировку** и выберите поле — внесенные изменения также отобразятся в режиме макета. В отчет можно добавить 10 уровней группировки и при необходимости можно выполнить сортировку для каждого из них.

Если отчет содержит не менее одной группировки, то в него можно добавить итоги.

Для добавления итогов в отчет необходимо:

1. Открыть отчет в режиме макета или конструктора.
2. На вкладке **Конструктор** в группе **Группировка и итоги** нажать кнопку **Группировка**. Под отчетом появится область **Группировка, сортировка и итоги**.
3. Выбрать уровень группировки, а затем нажать кнопку **Больше**.
4. Найти поле «с итогами» и щелкнуть стрелку рядом с ним.
5. В диалоговом окне **Итоги** выбрать поле, которое требуется вычислить, и применяемый тип вычисления (например, сумма или среднее значение), а также настроить параметры, такие как общий итог и промежуточные итоги.

В Access есть два базовых макета отчета — табличный и «в столбик».

Табличные макеты напоминают электронные таблицы. Используйте их, когда необходимо представить данные в простом формате списка.

Макеты «в столбик» похожи на формы, которые заполняют в банках или при совершении покупок через Интернет. Используйте такой тип макетов, если отчет содержит слишком много полей для отображения в табличном виде.

В смешанных макетах используются элементы и табличного макета, и макета «в столбик». Например, в пределах отчета часть полей может размещаться вдоль строки, а остальная часть — в столбик. В приложении Access 2010 смешанные макеты создаются вручную в режиме макета либо в режиме конструктора.

ЗАДАНИЕ № 1

Visual FoxPro 9.0

1. С использованием мастеров отчетов создать простой отчет для таблицы, хранящей сведения о студентах; отчет с группированием данных; отчет для связанных таблиц.
2. Создать отчет, пользуясь конструктором отчетов, включив в него все полосы. Выполнить группировку студентов по специальностям, группам и в алфавитном порядке внутри каждой из групп. Создать вычисляемые поля,

подсчитав количество студентов в каждой группе и общее количество студентов, обучающихся по каждой специальности.

В Microsoft Access 2010

1. С использованием мастеров отчетов создать простой отчет для таблицы, хранящей сведения о студентах; отчет с группированием данных; отчет для связанных таблиц.
2. Создать отчет, пользуясь конструктором отчетов, включив в него все полосы. Выполнить группировку студентов по специальностям, группам и в алфавитном порядке внутри каждой из групп. Создать вычисляемые поля, подсчитав количество студентов в каждой группе и общее количество студентов, обучающихся по каждой специальности.
3. Создать отчет, пользуясь конструктором отчетов, содержащей сведения об учебных планах, сгруппировав данные по группам и семестрам. В отчет включить данные о количестве дисциплин, изучаемых группой в каждом семестре.

Контрольные вопросы

1. Что Вы понимаете под отчетом?
2. Какие элементы включает отчет?
3. В чём преимущества группировки данных и как функционирует механизм группировки?
4. Как осуществляется создание вычисляемых полей?
5. Что такое шапка и резюме отчета?
6. Какие возможности имеет конструктор отчетов при проектировании оригинальных отчетов в Visual FoxPro 9.0?
7. Какие возможности имеет конструктор отчетов при проектировании оригинальных отчетов в Microsoft Access 2010?
8. Какие средства имеются в Visual FoxPro 9.0 для создания отчетов?
9. Какие средства имеются в Microsoft Access 2010 для создания отчетов?

ЛАБОРАТОРНАЯ РАБОТА № 4

РАЗРАБОТКА ЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ СРЕДСТВАМИ ERWIN. НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ

1. Разработать модель «Сущность - связь» с использованием ERWin по индивидуальному заданию и предварительно определить структуру таблиц базы данных.

2. Согласовать с преподавателем. Варианты индивидуальных заданий приведены в приложении 4.
3. Провести анализ полученных схем отношений на предмет нахождения их в 3 нормальной форме.
4. Создать схему базы данных в Visual FoxPro 9.0.

ЛАБОРАТОРНАЯ РАБОТА № 5

РАЗРАБОТКА ФИЗИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ НА ЯЗЫКЕ TRANSACT SQL, КОМАНДЫ ВЕДЕНИЯ ДАННЫХ

1. Используя язык Transact SQL (диалект языка SQL в СУБД MS SQL Server) и модель базы данных в соответствии с индивидуальным заданием предыдущей лабораторной работы создать схему взаимосвязанных отношений.
2. Построить диаграмму данных. Проверить её соответствие логической модели базы данных.
3. Заполнить записями каждое отношение, используя команду добавления записей.
4. Выполнить операции модификации записей, используя подзапрос. Выполнить изменение одновременно нескольких полей. Выполнить изменение записи, используя раздел FROM для расширения табличного пространства.
5. Выполнить удаление записей, удовлетворяющих условию. Выполнить удаление указанного количества записей. Выполнить удаление записей, используя подзапрос.

ЛАБОРАТОРНАЯ РАБОТА № 6

РЕАЛИЗАЦИЯ ЗАПРОСОВ НА ЯЗЫКЕ TRANSACT SQL

Используя язык Transact SQL (диалект языка SQL в СУБД MS SQL Server) и модель базы данных в соответствии с индивидуальным заданием создать следующие запросы:

1. Создать и реализовать запрос, используя скалярную функцию.
2. Создать и реализовать запросы, используя разделы Distinct и ALL.
3. Создать и реализовать запрос, используя переименование столбцов различными способами.
4. Создать и реализовать запросы, используя операторы IN, LIKE, BETWEEN, AND, OR.

5. Создать и реализовать запросы, используя сортировку по одному и нескольким полям с различным видом сортировки.
6. Создать и реализовать запрос, используя группировку с функцией агрегирования. Создать и реализовать запрос, используя одновременно с группировкой ограничение HAVING.
7. Создать и реализовать запросы, используя все виды соединения таблиц.
8. Создать и реализовать запросы, используя оператор CASE (два варианта синтаксиса).
9. Создать и реализовать запросы, используя подзапросы в списке выводимых значений, в разделе FROM, в разделе WHERE с использованием операторов IN и LIKE.
10. Создать и реализовать запрос, используя команду UNION.

ЛАБОРАТОРНАЯ РАБОТА № 7

РАБОТА С КУРСОРАМИ, ВИДАМИ, ПОЛЬЗОВАТЕЛЬСКИМИ ТИПАМИ, ПРАВИЛАМИ, ЗНАЧЕНИЯМИ ПО УМОЛЧАНИЮ НА ЯЗЫКЕ TRANSACT SQL

Используя язык Transact SQL (диалект языка SQL в СУБД MS SQL Server) и модель базы данных в соответствии с индивидуальным заданием выполнить следующие пункты:

1. Создать и отобразить все записи из динамического, ключевого и статического курсоров. Обосновать разницу отображения обновленных данных в различных видах курсоров.
2. Используя динамический курсор, сделать изменения и удаления записей в базе данных.
3. Создать пользовательский тип. Показать работу с ним при операциях ведения данных.
4. Создать два правила. Привязать правила к полю и пользовательскому типу. Показать работу правил при операциях ведения данных.
5. Создать значение по умолчанию. Привязать значение по умолчанию к полю. Показать работу значения по умолчанию при операциях добавления и изменения записей.
6. Создать виды: простой, составной и вид из вида. Отобразить данные из видов. Для составного вида переименовать поля и показать некоторые из них. Обосновать особенности опции WITH CHECK OPTION при работе с видами.

ЛАБОРАТОРНАЯ РАБОТА № 8

РАБОТА С ХРАНИМЫМИ ПРОЦЕДУРАМИ И ФУНКЦИЯМИ НА ЯЗЫКЕ TRANSACT SQL

Используя язык Transact SQL (диалект языка SQL в СУБД MS SQL Server) и модель базы данных в соответствии с индивидуальным заданием выполнить следующие пункты:

1. Создать и вызвать хранимую процедуру без параметров.
2. Создать и вызвать хранимую процедуру с входным, выходным формальным параметром и входным параметром, имеющим значение по умолчанию. Реализовать вызов процедуры, используя позиционный, ключевой, смешанный виды передачи параметров.
3. Создать и вызвать скалярную хранимую функцию.
4. Создать и вызвать функцию, возвращающую табличное значение.

ЛАБОРАТОРНАЯ РАБОТА № 9

РАБОТА С ТРИГГЕРАМИ НА ЯЗЫКЕ TRANSACT SQL

Используя язык Transact SQL (диалект языка SQL в СУБД MS SQL Server) и модель базы данных в соответствии с индивидуальным заданием выполнить следующие пункты:

1. Создать и проверить работу триггера, реализующего сложное правило.
2. Создать и проверить работу триггера, реализующего сложное значение по умолчанию.
3. Создать и проверить работу триггера, реализующего уникальность поля, не являющегося первичным ключом.
4. Создать и проверить работу триггера, реализующего ссылочную целостность restrict.
5. Создать и проверить работу триггера, реализующего ссылочную целостность cascade.

Приложение 1

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ СИСТЕМЫ VISUAL FOXPRO 9.0 и MICROSOFT OFFICE ACCESS 2010

VISUAL FOXPRO 9.0

Характеристики таблиц и индексных файлов

| | |
|---|---------------|
| Максимальное количество записей в файле | 1 млрд |
| Максимальный размер файла таблицы | 2 Гбайт |
| Максимальное количество символов в записи | 65 500 |
| Максимальное количество полей в записи | 255 |
| Максимальное количество одновременно открытых таблиц | 255 |
| Максимальное количество символов в поле таблицы | 254 |
| Максимальное число байт в индексном ключе некомпактного индекса | 100 |
| Максимальное число байт в индексном ключе компактного индекса | 240 |
| Число открытых индексных файлов таблицы | Не ограничено |
| Число открытых индексных файлов во всех рабочих областях | Не ограничено |
| Число устанавливаемых отношений | Не ограничено |
| Длина выражений, определяющих отношения | Не ограничена |

Характеристики полей

| | |
|---|----------------|
| Максимальный размер символьного поля | 254 |
| Максимальный размер числового поля | 20 |
| Максимальная длина имени поля свободной таблицы | 10 |
| Максимальная длина имени поля таблицы, содержащейся в базе данных | 128 |
| Минимальное значение целого числа | -2 147 483 647 |
| Максимальное значение целого числа | 2 147 483 647 |
| Точность вычислений | до 16 разрядов |

Характеристики переменных и массивов

| | |
|---|--------|
| Количество переменных по умолчанию | 1024 |
| Максимальное количество переменных | 65 000 |
| Максимальное количество массивов | 65 000 |
| Максимальное количество элементов в массиве | 65 000 |

Характеристики программных и процедурных файлов

| | |
|---|---------------|
| Число строк в исходном программном файле | Не ограничено |
| Максимальный размер скомпилированного программного модуля | 64 Кбайт |
| Количество процедур в файле | Не ограничено |
| Максимальное количество вложенных уровней оператора DO | 128 |
| Максимальное количество вложенных уровней оператора READ | 5 ? 10 |
| Максимальное количество передаваемых параметров | 27 |
| Максимальное количество вложенных команд при структурном программировании | 384 |
| Максимальное количество передаваемых параметров | 27 |
| Максимальное количество транзакций | 5 |

Характеристики Конструктора отчета

| | |
|---|---------------|
| Количество объектов, обрабатываемых отчетом | Не ограничено |
| Максимальная длина определения отчета | 20 дюймов |
| Максимальное количество уровней группирования | 128 |
| Максимальная длина символьных переменных отчета | 255 |

Другие характеристики

| | |
|---|--------------------------------------|
| Количество открытых окон различных типов | Не ограничено |
| Максимальное количество открытых BROWSE-окон | 255 |
| Максимальное число символов в символьной строке | 2 гигабайта |
| Максимальное количество символов в командной строке | 8 192 |
| Максимальное число символов в строке макроподстановки | 8 192 |
| Максимальное количество открытых файлов | Ограничивается операционной системой |
| Максимальное количество используемых клавиш, назначаемых в макросах | 1 024 |
| Максимальное количество полей, обрабатываемых командой SQL SELECT | 255 |

Стандартные расширения файлов

Для ориентирования в системе файлов Visual FoxPro 9.0, ниже приведены примеры стандартных расширений имен для различных типов файлов.

| <i>Стандартное расширение имени файла</i> | <i>Тип файла</i> |
|--|--|
| *.DBC | Файл базы данных |
| *.DCT | Мето-файл базы данных |
| *.DBF | Файл таблицы |
| *.FPT | Файл Мето-поля таблицы |
| *.IDX | Индексный файл |
| *.CDX | Мультииндексный файл |
| *.PRG | Программный файл |
| *.FXP | Скомпилированный программный файл |
| *.SCX/* .SCT | Файл описания экранной формы |
| *.SPX | Скомпилированная программа формы |
| *.SPR | Сгенерированная программа формы |
| *.MNX/* .MNT | Файл описания меню |
| *.MPX | Скомпилированная программа меню |
| *.MPR | Сгенерированная программа меню |
| *.QPR | Программа SQL– запросов |
| *.QPX | Скомпилированная программа SQL– запросов |
| *.FRX/* .FRT | Файл описания отчета |
| *.LBX/* .LBT | Файл описания этикеток |
| *.VUE | Файл представления |
| *.PJX/* .PJT | Файл описания проектов |
| *.APP | Сгенерированный файл приложения |
| *.VCX/* .VCT | Файл описания библиотеки классов |

MICROSOFT OFFICE ACCESS 2010

Основные общие характеристики

| | |
|--|--|
| Максимальное значение размера файла базы данных Microsoft Access (accdb) | 2 ГБ за вычетом места, необходимого системным объектам |
| Максимальное число объектов в базе данных | 32 768 |
| Максимальное число модулей (включая формы и отчеты, у которых свойство HasModule (наличие модуля) имеет значение True) | 1 000 |
| Максимальное число знаков в имени объекта | 64 |
| Максимальное число знаков в пароле | 20 |
| Максимальное число знаков в имени пользователя или имени группы | 20 |
| Максимальное число одновременно работающих пользователей | 255 |

Характеристики таблиц

| | |
|---|--|
| Максимальное число знаков в имени таблицы | 64 |
| Максимальное число знаков в имени поля | 64 |
| Максимальное число полей в таблице | 255 |
| Максимальное число открытых таблиц | 2048 (фактическое число может быть меньше из-за внутренних таблиц, открытых в Access) |
| Максимальный размер таблицы | 2 ГБ за вычетом места, необходимого системным объектам |
| Максимальное число знаков в текстовом поле | 255 |
| Максимальное число знаков в поле MEMO | 65 535 при вводе данных через интерфейс пользователя; 2 ГБ для хранения знаков при программном вводе данных |
| Максимальный размер поля объекта OLE | 1 ГБ |
| Максимальное число индексов в таблице | 32 |
| Максимальное число полей в индексе | 10 |
| Максимальное число знаков в сообщении об ошибке | 255 |
| Максимальное число знаков в условии на значение | 2 048 |
| Максимальное число знаков в описании таблицы или поля | 255 |
| Максимальное число знаков в записи (кроме полей MEMO и полей объектов OLE), если свойству Сжатие Юникод полей присвоено значение Да | 4 000 |
| Максимальное число знаков в значении свойства поля | 255 |

Основные характеристики запросов

| | |
|--|--|
| Максимальное число установленных связей | 32 на одну таблицу за вычетом числа индексов, находящихся в таблице для полей или сочетаний полей, не участвующих в связях |
| Максимальное число таблиц в запросе | 32 |
| Максимальное число объединений в запросе | 6 |
| Максимальное число полей в наборе записей | 255 |
| Максимальный размер набора записей | 1 ГБ |
| Предел сортировки | 255 знаков в одном или нескольких полях |
| Максимальное число уровней вложения запросов | 50 |
| Максимальное число знаков в ячейке на бланке запроса | 1 024 |
| Максимальное число знаков в параметре в запросе с параметрами | 255 |
| Максимальное число операторов AND в предложении WHERE или HAVING | 99 |
| Максимальное число знаков в инструкции SQL | приблизительно 64 000 |

Некоторые из максимальных значений могут быть меньше, если запрос содержит поля подстановок, одновременно допускающих несколько значений.

Основные характеристики форм и отчетов

| | |
|--|--|
| Максимальное число знаков в надписи | 2 048 |
| Максимальное число знаков в поле | 65 535 |
| Максимальная ширина формы или отчета | 22 дюйма (55,87 см) |
| Максимальная высота раздела | 22 дюйма (55,87 см) |
| Максимальная высота всех разделов вместе с заголовками разделов (в режиме конструктора) | 200 дюймов (508 см) |
| Максимальное число уровней вложения форм или отчетов | 7 |
| Максимальное число полей или выражений, которые можно отсортировать или сгруппировать в отчете | 10 |
| Максимальное число заголовков и примечаний в отчете | 1 заголовок/ примечание отчета; 1 заголовок/ примечание страницы; 10 заголовков/ примечаний групп |
| Максимальное число печатных страниц в отчете | 65 536 |
| Максимальное число элементов управления и разделов, которые можно добавить за время существования формы или отчета | 754 |
| Максимальное число знаков в инструкции SQL, используемой в качестве свойства RecordSource (источник записей) или RowSource (источник строк) формы, отчета или элемента управления (как ACCDB, так и ADP) | 32 750 |

Основные характеристики макросов

| | |
|--|-----|
| Максимальное число макрокоманд в макросе | 999 |
| Максимальное число знаков в условии | 255 |
| Максимальное число знаков в комментарии | 255 |
| Максимальное число знаков в аргументе макрокоманды | 255 |

Основные общие характеристики проектов

| | |
|--|---|
| Максимальное число объектов в проекте Access (ADP) | 32 768 |
| Максимальное число модулей (включая формы и отчеты, у которых свойство HasModule (наличие модуля) имеет значение True) | 1 000 |
| Максимальное число знаков в имени объекта | 64 |
| Максимальное число столбцов в таблице | 250 (Microsoft SQL Server 6.5); 1024 (Microsoft SQL Server 7.0, 2000 и 2005) |

Характеристики форм и отчетов проектов

| | |
|--|---|
| Максимальное число знаков в надписи | 2 048 |
| Максимальное число знаков в поле | 65 535 |
| Максимальная ширина формы или отчета | 22 дюйма (55,87 см) |
| Максимальная высота раздела | 22 дюйма (55,87 см) |
| Максимальная Высота всех разделов вместе с заголовками разделов (в режиме конструктора) | 200 дюймов (508 см) |
| Максимальное число уровней вложения форм или отчетов | 7 |
| Максимальное число полей или выражений, которые можно отсортировать или сгруппировать в отчете | 10 |
| Максимальное число заголовков и примечаний в отчете | 1 заголовок/примечание отчета; 1 заголовок/примечание страницы; 10 заголовков/примечаний групп |
| Максимальное число печатных страниц в отчете | 65 536 |
| Максимальное число элементов управления и разделов, которые можно добавить за время существования формы или отчета | 754 |
| Максимальное число знаков в инструкции SQL, используемой в качестве свойства RecordSource (источник записей) или RowSource (источник строк) формы, отчета или элемента управления (как ACCDB, так и ADP) | 32 750 |

Характеристики макросов проектов

| | |
|--|-----|
| Максимальное число макрокоманд в макросе | 999 |
| Максимальное число знаков в условии | 255 |
| Максимальное число знаков в комментарии | 255 |
| Максимальное число знаков в аргументе макрокоманды | 255 |

Приложение 2

ОСНОВНЫЕ ТИПЫ ПОЛЕЙ VISUAL FOXPRO 9.0.

| <i>Тип поля</i> | <i>Содержание</i> |
|---|---|
| Character (Символьный) | Любые символы. Прописные и строчные буквы обрабатываются по-разному. Максимальный размер символьного поля – 254 символа. |
| Numeric (Числовой) | Числа (целые или дробные), подлежащие математической обработке. Содержит числа от 0 до 9, а также знак (необязательно) и десятичную запятую (необязательно). Дробную часть числа вводить необязательно. Диапазон: от $-.9999999999E+19$ до $.9999999999E+20$. |
| Integer (Числовой, целочисленный) | Целые числа. Хранятся в таблицах как четырехбайтные двоичные значения и поэтому занимают меньше памяти, чем другие числовые типы данных, так как двоичные значения не требуют преобразования в ASCII. Диапазон от -2147483647 до 2147483646 . |
| Float (Вещественный) | Числа, имеющие дробную часть (представляются в формате с плавающей точкой), определенные с обычной точностью. |
| Double (Вещественный с двойной точностью) | Числа, имеющие дробную часть (представляются в формате с плавающей точкой), определенные с двойной точностью. Если необходима большая точность, фиксированный объем памяти в таблице или значения с плавающей запятой, то лучше использовать этот тип вместо Numeric. Диапазон от $+/-4.9406564584124E-324$ до $+/-1.79769313486232E+308$. |
| Currency (Денежный) | Поле для представления денежных сумм. Используется для хранения чисел с четырьмя знаками после запятой. Если вы укажете более четырех десятичных цифр после десятичной запятой, то Visual FoxPro автоматически округлит их до четырех. Диапазон от $-922337203685477,5808$ до $922337203685477,5807$. |
| Date (Дата) | Дата, определяющая день, месяц и год. Ввод в это поле контролируется автоматически. При вводе значения в поле по умолчанию принят формат даты – мм/дд/гг. Для изменения формата записи даты, например, на дд/мм/гг, необходимо ввести команду – Set Date British. |
| DateTime (Дата и время) | Дата и время. Формат вводимого значения в поле базы данных по умолчанию имеет вид – мм/дд/гг чч:мм:сс. |
| Logical (Логический) | Логические данные. Используется для хранения информации, принимающей только два значения – «истина» (.T.) или «ложь» (.F.). Занимает в памяти 1 байт. |

| | |
|---|---|
| Мемо (Текстовое поле произвольной длины) | Примечания, размер которых неограничен. Могут содержать любую символьную информацию. Фактически это указатель на блок данных в файле с расширением .FPT (файлы .FPT и .DBF одноименны). Доступ к файлу .FPT возможен только при открытом одноименном файле .DBF, поэтому при перемещении информации с одного носителя на другой необходимо копировать оба файла. Для ввода значения в поле нужно два раза щёлкнуть мышью по полю мемо, после чего в открывшемся окне редактирования набрать нужный текст примечаний. Закрыв окно редактирования, значение поля примечаний будет сохранено, а в поле базы данных слово мемо будет заменено на Мемо, что является признаком его заполнения. |
| General (Двоичное поле произвольной длины) | Поле, в котором хранится и отображается OLE-объект (Object Linking and Embedding), созданный другими приложениями. OLE-объект может содержать графику, звук, видеоклип, документы Word for Windows, таблицы Microsoft Excel и т.п. Для ввода в поле базы данных объекта нужно выполнить двойной щелчок мыши на поле gen, после чего появится окно редактирования и изменится меню, затем выбрать пункт меню Edit-Insert Object и из перечня приложений, поддерживающих технологию OLE, выбрать нужное Вам (например рисунок Paintbrush). Закрыв окно приложения и окно редактирования поля, в базе данных поле gen будет заменено на Gen, что является признаком его заполнения. |
| Character binary | Используется для хранения двоичного текста длиной до 254 символов. Содержит любые символьные данные, которые вы хотите использовать, не изменяя кодовую страницу. |
| Мемо binary | Поле примечаний для хранения двоичного текста произвольной длины. Содержит любые данные поля мемо, которые вы хотите использовать, не изменяя кодовую страницу. |

Приложение 3

ОСНОВНЫЕ ТИПЫ ДАННЫХ MICROSOFT ACCESS 2010

Текстовый (Text) — символьные или числовые данные, не требующие вычислений. Поле данного типа может содержать до 255 символов. Размер текстового поля задается с помощью свойства Размер поля (FieldSize), в котором указывается максимальное количество символов, которые могут быть введены в данное поле. При необходимости ввести строку большего размера, чем указано в данном свойстве, надо значение свойства изменить, т. к. место под дополнительные символы не резервируется. Независимо от длины текстового поля оно хранится в базе данных Access в записи переменной длины. Access не занимает память под неиспользуемую часть текстового поля в записи, что уменьшает общий размер базы данных.

Поле MEMO (MEMO) — поле MEMO предназначено для ввода текстовой информации, по объему превышающей 255 символов. Такое поле может содержать до 65 535 символов. Этот тип данных отличается от типа Текстовый (Text) тем, что в таблице хранятся не сами данные, а ссылки на блоки данных, хранящиеся отдельно. За счет этого ускоряется обработка таблиц (сортировка, поиск и т. п.). Поле типа MEMO не может быть ключевым или проиндексированным. Не используйте поле типа MEMO, если нужно будет сортировать записи таблицы по данному полю — это невозможно. Если 255 символов не достаточно для хранения всех данных, придется разбить его на два или три текстовых поля.

Ввод данных в это поле можно выполнить непосредственно в поле таблицы, либо через область ввода, вызываемую нажатием комбинации клавиш <Shift>+<F2>.

Числовой (Number) — числовой тип применяется для хранения числовых данных, используемых в математических расчетах. Имеет много подтипов. От выбора подтипа (размера) данных числового типа зависит точность вычислений. Для установки подтипа числовых данных служит свойство Размер поля (FieldSize). Данные этого типа могут содержаться в 1, 2, 4, 8 или 16 байтах. Обычно по умолчанию используется подтип Длинное целое (LongInteger), который занимает 4 байта и представляет собой число в пределах от -2 147 483 648 до +2 147 483 647. Но, кроме этого типа, можно указать Байт (Byte) — 1 байт, Целое (Integer) — 2 байта, Одинарное с плавающей точкой (Single) — 4 байта, Двойное с плавающей точкой (Double) — 8 байтов, Десятичное (Decimal) — 12 байтов, Код репликации (Replication ID) — 16 байтов.

Дата/Время (Date/Time) — тип для представления даты и времени. Позволяет вводить даты с 100 по 9999 год. Размер поля — 8 байтов. Даты и время хранятся в специальном фиксированном числовом формате. Дата является целой частью значения поля типа Дата/Время, а время — его дробной частью. Access предоставляет большой выбор форматов отображения даты и времени.

Денежный (Currency) — тип данных, предназначенный для хранения данных, точность представления которых колеблется от 1 до 4 десятичных знаков. Целая часть данного типа может содержать до 15 десятичных знаков.

Счетчик (AutoNumber) — поле содержит 4-байтный уникальный номер, определяемый Microsoft Access автоматически для каждой новой записи либо случайным образом, либо путем увеличения предыдущего значения на 1. Значения полей типа счетчика обновлять нельзя. Максимальное число записей в таблице с полем счетчика не должно превышать двух миллиардов.

Логический (Yes/No) — логическое поле, которое может содержать только два значения, интерпретируемых как Да/Нет, Истина/Ложь, Включено/Выключено. Поля

логического типа не могут быть ключевыми, но их можно индексировать. Access использует величину -1 для представления значения Истина и величину 0 — для значения Ложь.

Поле объекта OLE (OLE object) — содержит ссылку на OLE-объект (лист Microsoft Excel, документ Microsoft Word, звук, рисунок и т. п.). Объем объекта ограничивается имеющимся в наличии дисковым пространством. Тип объекта OLE не указывается в свойствах поля объекта OLE. Он зависит от приложения, использованного для создания объектов OLE, которые хранятся в этом поле. Упаковщик объектов позволяет внедрять файлы, созданные приложениями, которые не являются серверами объектов OLE. Например, он позволяет внедрить текстовые файлы в качестве объектов в таблицы Access 2002. В поле объекта OLE могут храниться произвольные данные, в том числе и данные нескольких типов. Это позволяет обойти основное ограничение реляционных баз данных, которое требует, чтобы в каждом поле хранились данные только одного типа.

Гиперссылка (Hyperlink) — дает возможность хранить в поле ссылку, с помощью которой можно сослаться на произвольный фрагмент данных внутри файла или Web-страницы на том же компьютере, в интранет или в Интернет. Гиперссылка состоит из четырех частей: отображаемый текст, адрес (путь к файлу или странице), дополнительный адрес (положение внутри файла или страницы) и текст всплывающей подсказки. Каждая часть гиперссылки может содержать до 2048 символов. Поле типа Гиперссылка не может быть ключевым или индексированным.

Приложение 4

ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ

Вариант № 1

БД промышленных предприятий

В администрации области ведется учет промышленных предприятий области по различным отраслям народного хозяйства в соответствии с общесоюзным классификатором отраслей народного хозяйства (ОКОНХ). О каждом предприятии фиксируется следующая информация: ИНН, статистический регистрационный код по общероссийскому классификатору предприятий и организаций (ОКПО), название предприятия, численность сотрудников, юр. адрес, телефон, e-mail, ФИО руководителя, а также ведется перечень выпускаемой продукции по годам с указанием наименования продукции (в соответствии с общероссийским классификатором продукции ОКП), объема (планируемого и фактического) и себестоимости выпускаемой продукции. Одинаковую продукцию могут выпускать несколько предприятий, а предприятие может выпускать несколько наименований продукции. Кроме того ведется информация о взаимной кооперации промышленных предприятий, кто является поставщиком или потребителем того или иного продукта. Одно и то же предприятие может быть и поставщиком и потребителем различной продукции. Учет объема взаимной кооперации предприятий по различным видам продукции ведется по годам. Объем выпуска продукции и объем взаимной кооперации осуществляется в одинаковых единицах измерений (код единицы измерений, наименование единицы измерений).

Вариант № 2

БД программного обеспечения (ПО) в университете

В университете ведется учет имеющегося ПО и использования его на различных кафедрах университета. Учет ведется по видам ПО (код вида, название вида ПО). Например, системное (которое в свою очередь делится на операционные системы, встроенные программы, утилиты, системы программирования, СУБД), прикладное, инструментальное и т.д.

По каждому ПО имеются следующие данные: название, № версии, фирма разработчик (код фирмы разработчика, название фирмы), количество лицензий, дата приобретения.

На кафедры (название кафедры, ФИО зав. кафедрой) ПО передается ответственному за ПО. На одной кафедре может быть несколько ответственных за ПО. О каждом ответственном за ПО хранятся следующие данные: ФИО, должность, дата назначения, дата снятия полномочий (может отсутствовать). При передаче ПО на кафедры фиксируется дата передачи, название кафедры, название ПО и № версии, количество экземпляров, если по количеству экземпляров существует ограничение.

Вариант № 3

БД испытаний новых изделий

На предприятии проводятся испытания новых изделий. Каждое изделие имеет название, шифр чертежа и уникальный номер и может испытываться неоднократно разными испытателями (код испытателя, ФИО, должность), в том числе в течение одних суток. С изделием определённого шифра чертежа связан фиксированный набор параметров определённой размерности, измеряемых в процессе испытания. Часть параметров измеряется многократно, для определённости запоминаются значение и

порядковый номер измерения в данном испытании. Параметры могут выходить за пределы нормы, задаваемой числовым диапазоном для изделия определённого шифра чертежа. Наборы измеряемых параметров для изделий разных шифров могут пересекаться по названиям.

Вариант № 4

БД пенсионного фонда

Районный пенсионный фонд ведет учет денежных средств, поступающих от предприятий района. Каждая организация предоставляет следующую информацию: регистрационный номер в пенсионном фонде (уникальный), индекс ИНН (уникальный), название, форма собственности, ФИО директора, ФИО главного бухгалтера, а также список сотрудников данного предприятия:

- № страхового свидетельства
- ФИО сотрудника
- дата приема
- дата увольнения (может отсутствовать).

Дополнительно каждый месяц предприятия предоставляют в пенсионный фонд сведения о заработной плате сотрудников:

- № страхового свидетельства,
- ФИО сотрудника,
- Месяц
- Год
- сумма заработной платы.

Организация осуществляет перечисления в пенсионный фонд: № платежного поручения, дата, общая сумма перечислений. В назначении платежа указывается сумма и месяц и год, за который производится перечисление денег. По одному платежному поручению могут быть осуществлены перечисления по нескольким отчетным периодам.

Вариант № 5

БД особо ответственных деталей изделий авиапредприятия

На предприятии, занимающемся производством авиационной техники, осуществляется контроль изготовления особо ответственных деталей выпускаемых изделий. Для каждого изделия (код изделия, наименование изделия) гл. конструктором составляется перечень особо ответственных деталей (номер детали по конструкторской документации, наименование детали по конструкторской документации). Номера деталей по конструкторской документации для различных изделий могут повторяться. Для каждой такой детали технолог определяет перечень особо ответственных технологических операций (ОТО) (номер технологической операции и наименование технологической операции) и указывается подразделение (номер подразделения, название подразделения), где будет выполняться данная технологическая операция. После выполнения технологической операции производится контроль качества ее выполнения: фиксируется дата выполнения ОТО, количество деталей без брака, Количество деталей с браком. В случае выявления брака фиксируются все выявленные несоответствия (код несоответствия, наименование несоответствия) и их количество.

Вариант № 6

БД курсов повышения квалификации

В университете ведется учет прохождения курсов повышения квалификации сотрудниками университета. При этом, в зависимости от занимаемой должности (код

должности, название должности), существуют правила, устанавливающие порядок прохождения курсов (например, 1 раз в пять лет). Информацию о курсах на кафедры (название кафедры, ФИО зав. кафедрой) предоставляет учебный отдел в зависимости от перечня специальностей (шифр специальности, название специальности) по которым готовит кафедра. При этом указываются название курсов, организация, место проведения, сроки проведения, количество часов, список специальностей для которых предназначены данные курсы. Запись на курсы производится на кафедрах секретарем кафедры, и список передается в учебный отдел. Информацию о прохождении курсов с указанием ФИО, должности и полученного регистрационного № свидетельства секретарь кафедры передает в учебный отдел.

Вариант № 7

БД результатов тестирования студентов

Деканат ведет учет тестирования студентов по различным дисциплинам. Все тестирование, которое проходят студенты, делится на виды, например, текущее, министерское. Тестирование осуществляется либо по какой-то дисциплине (код дисциплины, название дисциплины) либо по конкретному разделу дисциплины (код раздела, наименование раздела). Тестирование может проводиться как для студентов одной группы (№ группы, курс, кол-во студентов), так и нескольких групп одновременно. О каждом тестировании фиксируется следующая информация: дата, ФИО преподавателя, проводящего тестирование, дисциплина или раздел дисциплины, общее количество вопросов, необходимое для прохождения теста количество верных ответов, а также результаты тестирования для каждого студента: количество верных и неверных ответов.

Вариант № 8

БД лицензий на выполнение работ на авиапредприятии

Авиапредприятие занимается производством, ремонтом и обслуживанием различных типов летательных аппаратов (ЛА) (код типа, название типа ЛА).

Выполнять работы, связанные с производством, ремонтом и обслуживанием ЛА могут только квалифицированные специалисты, имеющие определенную специальность по общероссийскому классификатору специальностей по образованию (ОКСО). Кроме того каждый специалист обязан пройти обучение в специально аккредитованных организациях (ИНН, название организации, адрес, телефон, e-mail, № и дата аттестата аккредитации) и получить лицензию на выполнение того или иного вида работ на указанных типах ЛА. В лицензии указывается № лицензии, наименование организации выдавшей лицензию, период обучения, срок действия лицензии, типы ЛА и перечень видов работ, который может выполнять специалист согласно этой лицензии.

Вариант № 9

БД «Кадровый учет»

Отдел кадров ведет учет сотрудников, работающих на предприятии. Каждый сотрудник характеризуется уникальным табельным номером, ФИО, полом, датой рождения, домашним адресом и может иметь домашний и сотовый телефон. Сотрудник может иметь детей. Каждый ребенок характеризуется ФИО, полом и датой рождения.

Каждый сотрудник в конкретный момент времени может работать только в одном из подразделений предприятия (код подразделения, наименование подразделения) в определенной должности (код должности, наименование должности), но в течение всей трудовой деятельности на данном предприятии он может неоднократно переходить из одного подразделения в другое, с одной должности на другую.

Все кадровые операции (прием, увольнение, перевод на другую должность, в другое подразделение и т.д.) осуществляются на основании соответствующего приказа (дата и № приказа). Дата и содержание операции учитываются в личной карточке сотрудника.

Вариант № 10

БД строительных конструкций

В цехах предприятия стройиндустрии изготавливаются детали для сборки строительных конструкций. Детали имеют название и шифр чертежа детали. Каждый чертеж кроме шифра имеет название и ФИО разработчика. Детали одного названия могут изготавливаться по нескольким чертежам. В состав конструкции может входить несколько деталей определённого чертежа. Сборка конструкций производится бригадами. Каждая бригада имеет уникальный номер, возглавляется бригадиром и за смену собирает определенное количество различных конструкций. Конструкция имеет название, шифр чертежа конструкции и уникальный номер. Конструкция одного названия может собираться по различным чертежам.

Вариант № 11

БД выбросов предприятий

Экологическая служба города ведет учет выбросов в атмосферу дымовых газов различными предприятиями. Каждое предприятие характеризуется кодом ОКПО (уникальным), названием, ФИО руководителя, районом, адресом, выброс – количеством кубометров газа, датой и временем начала и окончания выброса, а также составом примесей в процентах.

Примеси делятся на классы (код класса, название класса, например, твердые, жидкие и газообразные) и характеризуются в выбросе названием, максимальным размером частиц и средним весовым содержанием в одном кубометре. Одна и та же примесь может относиться к различным классам.

Вариант № 12

БД нефтяных месторождений

Федеральное агентство по недропользованию ведет учет данных по имеющимся нефтяным месторождениям и их основным технико-экономическим показателям.

Месторождение характеризуется названием (уникальное) и месторасположением (округ, регион, район).

По каждому месторождению хранятся данные об основных технико-экономических показателях (например, таких как добыча нефти, товарная нефть валовая продукция, обводненность нефти (весовая), строительство скважин, коэффициент эксплуатации действующего фонда нефтяных скважин) по годам и план добычи нефти. Каждый технико-экономический показатель характеризуется кодом показателя, названием показателя и измеряется в определенных единицах измерения (код единицы измерения, название единицы измерения).

Вариант № 13

БД книг в библиотеке

Библиографический отдел библиотеки ведет учет новых поступлений книг. Каждое издание книги характеризуется:

— одним или несколькими авторами (могут отсутствовать);

- названием;
- местом издания;
- издательством;
- годом издания;
- типом издания (художественная литература, учебники, научно-техническая литература и т.п.);
- ISBN (международный стандартный книжный номер) — уникальный номер книжного издания, необходимый для распространения книги в торговых сетях и автоматизации работы с изданием.

Издание может поступать в нескольких экземплярах, каждому из которых присваивается уникальный инвентаризационный номер. Каждый экземпляр хранится в определенном отделе библиотеки (код отдела, название отдела). Некоторые экземпляры помечаются как особо ценные (контрольные).

Вариант № 14

БД музеев

Министерство культуры города ведет учет музеев (код музея, название музея, адрес, ФИО руководителя, номер телефона) и экспонатов, имеющих в музеях города. Классификация музеев осуществляется по профилю музеев. Например, выделяются следующие группы музеев:

1. Исторические музеи:
 - а) общеисторические (вся история человечества);
 - б) национальной истории (история отдельных стран);
 - в) региональной истории (история отдельных регионов);
 - г) истории отдельных периодов или исторических событий, феноменов (историко-революционные, истории декабризма, истории религии или атеизма и т.п.);
 - д) узкоспециализированные (археологические, этнографические, нумизматические, военно-исторические и т.д.).
2. Музеи отдельных отраслей культуры (в узком смысле слова):
 - а) художественные музеи;
 - б) литературные музеи;
 - в) музыкальные музеи;
 - г) театральные музеи;
 - д) музеи физкультуры и спорта.
3. Педагогические музеи.
4. Естественнонаучные музеи.
5. Промышленные музеи:
 - а) музеи торговли, промышленности и ремесел;
 - б) музеи техники (музеи отдельных отраслей);
 - в) музеи отдельных предприятий.
6. Сельскохозяйственные музеи.
7. Комплексные.

Каждый экспонат имеет уникальный номер, название, ФИО автора (может отсутствовать), век, год создания (может быть неизвестен) и относится к определенной группе (код группы, название группы). Экспонат может быть представлен в постоянной экспозиции музея или находится в запасниках.

Кроме того в музее могут быть организованы выставки. О каждой выставке должна храниться следующая информация: название выставки, организатор выставки, сроки проведения выставки, перечень экспонатов, представленных на данной выставке, с указанием из каких музеев они предоставлены.

Вариант № 15

БД продавцов компьютерной техники

В городской администрации хранится информация о фирмах, занимающихся распространением компьютерной техники. Каждая фирма характеризуется ИНН, названием, телефоном, факсом, адресом, ФИО директора. Кроме этого каждая фирма характеризуется формой собственности (государственная, ТОО, частная, акционерная, городская), общей площадью помещений, общим числом сотрудников.

Каждая фирма имеет штат сотрудников. О каждом сотруднике хранится следующая информация: ИНН, ФИО, дата поступления, дата увольнения (может отсутствовать). Один и тот же человек может работать одновременно в нескольких фирмах.

Весь товар относится к определенной группе товаров (код группы, наименование группы), которая в свою очередь может иметь неограниченное количество подгрупп.

У каждой фирмы есть прайс-лист по имеющемуся в продаже товару. В прайс-листе указывается: дата, группа товара, шифр товара, наименование, цена, срок гарантии.

Вариант № 16

БД профсоюзной организации

В профсоюзной организации хранится информация о членах профсоюза (номер профсоюзного билета, ФИО, стаж работы, дата вступления в профсоюз), а также о путевках и материальной помощи, которыми пользовались члены профсоюзной организации.

О каждой путевке хранится следующая информация:

- тип (лечебная, туристическая);
- наименование организации;
- город;
- срок действия (24 дня, 12 дней);
- № путевки (уникальный);
- дата, с которой действует данная путевка.

Путевки и материальная помощь выдаются членам профсоюза на основании заявления. В заявлении на путевку указывается ФИО, дата, организация, город, дата путевки. В заявлении на материальную помощь указывается: ФИО члена профсоюза, дата основания для оказания материальной помощи (одно или несколько). При регистрации заявления ему присваивается уникальный номер в пределах года и указывается дата регистрации.

О выдаче материальной помощи хранится следующая информация: дата выдачи, сумма, основание для оказания материальной помощи (код основания, название основания) и № заявления, на основании которого была выдана материальная помощь.

При выдаче путевки фиксируется № путевки, № заявления, на основании которого была выдана путевка.

В случае отказа фиксируется дата отказа, основание для отказа (код основания, название основания) и по какому заявлению.

Вариант № 17

БД справочной службы аптек

Справочная служба аптек ведет базу данных, в которой хранятся сведения обо всех аптеках города, товарах, продаваемых в аптеках и ценах на них. Аптека характеризуется номером, названием, телефоном, адресом, ФИО директора. Товар характеризуется кодом, наименованием и относится к определенной группе товаров (код группы, наименование группы), например, лекарственные препараты, сопутствующие товары и т. д. Каждая

группа товаров имеет определенный набор характеристик (код характеристики, наименование характеристики), которым описываются все товары, входящие в данную группу. Например, для лекарственных препаратов, это может быть активное действующее вещество, форма выпуска, дозировка, количество штук в упаковке, страна производитель и т.д. Товар с одинаковым названием, но различными значениями характеристик имеет различный код. Один и тот же товар может продаваться разными аптеками по разной цене и по различным условиям оплаты (наличные, безналичные).

Вариант № 18

БД поликлиники

Регистратура ведет учет пациентов, посещающих поликлинику. Пациент характеризуется № страхового полюса, ФИО, домашним адресом, полом и возрастом. Врач характеризуется ФИО, лечебным профилем (терапевт, хирург и т.п.) и имеет расписание (дата, время начала приема, время окончания приема, № кабинета). Для всех врачей одного профиля устанавливаются определенные нормы по количеству времени на прием одного пациента. Пациенты назначаются к врачу соответствующего профиля по расписанию его работы на определенное время. Во время приема пациентов врач фиксирует поставленный диагноз (код заболевания, название заболевания) В результате приема может быть зафиксировано несколько заболеваний, но заболевания могут и отсутствовать.

Вариант № 19

БД программного обеспечения (ПО) кафедры

На кафедре ведется учет имеющегося ПО. Учет ведется по видам ПО (код вида, название вида ПО). Например, системное (которое в свою очередь делится на операционные системы, встроенные программы, утилиты, системы программирования, СУБД), прикладное, инструментальное и т.д.

По каждому ПО имеются следующие данные: название, № версии, фирма разработчик (код фирмы разработчика, название фирмы), количество лицензий, дата поступления на кафедру, срок действия лицензии, а также для каких дисциплин используется данное ПО. Одно и то же ПО может быть использовано при изучении нескольких дисциплин (код дисциплины, название дисциплины). Кроме того хранятся требования к характеристикам компьютера (код характеристики, название характеристики, значение), на который может быть произведена установка данного ПО.

Вариант № 20

БД объектов строительной компании

Строительная компания ведет учет объектов строительства и перечень необходимых для каждого объекта материалов.

Каждый объект характеризуется уникальным кодом, адресом, сроком начала строительства, сроком сдачи, стоимостью и относится к определенному типу (например, производственное помещение, жилое здание, гараж и т.п.). Строительство объектов производится в соответствии с графиком производства работ, в котором указывается код этапа, название этапа, дата начала и дата окончания, стоимость этапа.

Материал характеризуется уникальным кодом, названием и измеряется в определенных единицах измерения (код единицы измерения, название единицы измерения).

Учет необходимых материалов ведется по этапам строительства, при этом указывается этап, название материала и его количество.

Вариант № 21

БД предприятия по техническому обслуживанию оборудования

Все оборудование, имеющееся на предприятии, подразделяется на группы (например, металлообрабатывающее, кузнечно-прессовое, сборочные автоматы и т.д.). Внутри каждой группы оборудование разделяется на подгруппы по технологическим признакам. Так, к примеру, металлообрабатывающее оборудование подразделяется на:

- токарные станки;
- сверлильные и расточные станки;
- шлифовальные и доводочные станки;
- станки для электро-физико-химической обработки;
- зубо- и резьбообрабатывающие станки;
- фрезерные станки;
- строгальные, долбежные и протяжные станки;
- разрезные станки.

Основными видами технического обслуживания являются:

- профилактический осмотр;
- планово-предупредительный ремонт;
- капитальный ремонт;
- малый ремонт;
- средний ремонт;
- пуско-наладочные работы;
- внеплановый ремонт.

Для каждой группы оборудования определен свой перечень видов технического обслуживания, с указанием периодичности проведения того или иного вида технического обслуживания, и перечень обязательных видов работ, которые должны быть выполнены при проведении данного вида технического обслуживания.

На каждую единицу имеющегося, или вновь поступившего оборудования составляется карточка, в которой указываются следующие данные:

- инвентарный номер;
- завод производитель;
- модель;
- краткое наименование;
- серийный номер (заводской);
- место установки (цех, подразделение);
- гарантийный срок;
- дата приемки;
- дата ввода в эксплуатацию.

Техническое обслуживание оборудования начинается с момента ввода его в эксплуатацию, т.е. с момента установки оборудования в том или ином подразделении предприятия.

В начале года на каждую инвентарную единицу составляются графики по видам технического обслуживания, в котором устанавливаются даты и сроки выполнения работ по данному оборудованию. Если какого-то вида обслуживания не запланировано, то соответствующий график не составляется. После выполнения работ по техническому обслуживанию ставится фактическая дата выполнения данных работ.

Вариант № 22

БД крови в пунктах переливания крови и доноров области

Министерство здравоохранения области ведет учет пунктов переливания крови, имеющегося там оборудования и крови, а также ведет базу данных доноров.

О каждом пункте переливания крови имеется следующая информация: № пункта, город, адрес, телефон, e-mail, ФИО заведующего пунктом, а также информация о имеющемся оборудовании (код оборудования, наименование оборудования) с указанием количества оборудования.

Учет крови ведется в соответствии с утвержденной номенклатурой крови (например, кровь цельная, кровь цельная лейкофильтрованная, кровь цельная лейкофильтрованная, для обменного переливания) ее компонентов (например, эритроцитная масса, эритроцитная масса облученная) и препаратов крови (например, раствор альбумина 5 %, иммуноглобулин антирезус). О каждом доноре хранится следующая информация: паспортные данные, адрес, ФИО, дата рождения, группа крови, резус-фактор, пол и присвоенный уникальный код донора.

В пунктах переливания крови каждой дачи (донации) крови и ее компонентов присваивается уникальный код дачи (донации) и фиксируются дата донации, данные о доноре, количество полученной крови или ее компонентов.

Вариант № 23

БД научно-исследовательского института (НИИ)

В НИИ производятся работы по различным темам. Каждая тема имеет уникальный шифр и наименование, руководителя темы (ФИО, должность, телефон), дату и номер договора, по которому выполняется данная тема, срок сдачи и общую сумму по договору. Каждая тема включает в себя определенное количество работ (код работы, наименование работы), каждая из которых может выполняться различными подразделениями НИИ, в зависимости от вида работ. Каждое подразделение характеризуется номером, наименованием, ФИО руководителя и может выполнять несколько видов работ. Каждое подразделение может одновременно выполнять работы по нескольким темам. Для выполнения работ по темам составляется график работ, в котором указывается код работы, наименование работы, подразделение, которое будет выполнять данную работу, планируемые сроки начала и окончания данных работ, а затем фиксируются фактические даты начала и окончания этих работ.

Вариант № 24

БД автосервиса

В автосервисе ведется учет автомобилей, которые были обслужены в данном автосервисе, а также перечень ремонтных работ, которые были проведены для данного автомобиля и учет используемых при ремонте запчастей.

Каждый автомобиль характеризуется гос. номером, маркой, годом выпуска и ФИО владельца.

Для выполнения ремонтных работ оформляется договор. В договоре указывается № и дата договора, дата окончания работ, ФИО заказчика, гос. номер автомобиля, а также перечень работ (код работы, наименование работы, количество, стоимость) с указанием необходимых для данной работы запчастей (шифр запчасти, наименование запчасти, количество, стоимость).

Оплата работ и запчастей производится в соответствии с прайс-листами на работы и запчасти. Стоимость работ и запчастей может меняться с течением времени.

Вариант № 25

БД спорткомитета

Спортивный комитет ведет учет командных соревнований, по различным видам спорта и результатов этих соревнований. Каждое соревнование характеризуется названием, рангом соревнования (городские, областные, международные и т.д.), страной и городом проведения, а также сроками проведения, количеством команд. Соревнование может проводиться как по одному, так и нескольким видам спорта. Каждая команды, принимающая участие в соревновании характеризуется названием, страной, городом, ФИО гл. тренера и составом команд (код спортсмена, ФИО). Результат соревнований фиксируется в следующем виде: соревнование, команда, место.

Вариант № 26

БД абонементного отдела библиотеки университета

Отдел абонемента производит запись читателей на абонемент и осуществляет библиотечное обслуживание.

Все читатели подразделяются на студентов и сотрудников. О студентах хранится следующая информация: № читательского билета, ФИО, факультет, № группы. О сотрудниках хранится следующая информация: № читательского билета, ФИО, факультет, подразделение университета, должность.

Каждая книга, имеющаяся в отделе, характеризуется:

- одним или несколькими авторами (могут отсутствовать);
- названием;
- изданием.

Книга может быть в нескольких экземплярах, каждому из которых присваивается уникальный инвентаризационный номер. Отдельные экземпляры могут быть помечены как контрольные.

При выдаче книг фиксируется № читательского билета, инвентаризационный номер, дата выдачи, срок возврата.

При возврате фиксируется фактическая дата возврата книги.

Вариант № 27

БД образовательных учреждений области

Министерство образования области ведет учет всех образовательных учреждений области. Все образовательные учреждения подразделяются на виды, например, государственные (федеральные или находящиеся в ведении субъекта РФ), муниципальные, негосударственные (частные; учреждения общественных и религиозных организаций (объединений)) и относится к определенному типу образовательного учреждения, который определяется в соответствии с уровнем и направленностью реализуемых им образовательных программ (дошкольные, общеобразовательные (начального общего, основного общего, среднего (полного) общего образования), начального профессионального образования, среднего профессионального образования, высшего профессионального образования, послевузовского профессионального образования и т.д.). Вид и тип образовательного учреждения определяет перечень разрешенных видов деятельности (код деятельности, наименование деятельности).

Каждое образовательное учреждение характеризуется: названием, датой создания, ФИО руководителя, городом, адресом, телефоном, и имеет утвержденный перечень разрешенных видов деятельности. Кроме того в базе данных хранится информация о выданных образовательному учреждению лицензиях на осуществление своей деятельности (номер, дата, срок действия лицензии, наименование организации, выдавшей лицензию).

Вариант № 28

БД результатов сдачи сессии

В деканате имеются сведения о группах студентов (номер группы, специальность), студентах (№ зачетки, ФИО, № группы), сведения об учебных планах и преподавателях (код преподавателя, ФИО, должность),

Каждую сессии (зимняя/весенняя, учебный год) деканат на основании учебных планов для каждой группы формирует список дисциплин (список может быть одинаковым для нескольких групп), по которым будет проводиться аттестация студентов и составляет график проведения аттестации. В графике указывается дата, номер группы, наименование дисциплины, ФИО преподавателя, вид аттестации.

Результаты аттестации фиксируются в ведомости в которой указывается дата, наименование дисциплины, ФИО преподавателя, номер зачетки, ФИО студента, результат аттестации.

Вариант № 29

БД платных медицинских услуг

Управление здравоохранения хранит сведения обо всех медицинских учреждениях города и перечне платных услуг, оказываемых в них.

Медицинское учреждение характеризуется ИНН, названием, телефоном, адресом, ФИО главврача. Услуги характеризуются следующими признаками: шифр, наименование, стоимость услуги и входят в определенную группу (код группы, название группы). При этом шифры услуг являются едиными для всех медучреждений, а стоимость может варьироваться в зависимости от медучреждения и даты оказания услуг.

При оказании услуги фиксируется № договора, дата договора, ФИО пациента и его паспортные данные, а также перечень оказанных услуг и их стоимость.

Вариант № 30

БД по разработке программного обеспечения (ПО)

Фирма, занимающаяся разработкой ПО, имеет штат сотрудников, имеющих определенные должности (код должности, наименование должности), согласно штатному расписанию. О каждом сотруднике хранятся следующие данные: код сотрудника, ФИО, должность.

Каждое ПО (шифр, название, вид ПО, стоимость, дата выпуска) разрабатывается группой разработчиков. Один и тот же разработчик, при разработке разного ПО, может быть использован в различных ролях (например, руководитель разработки, системный аналитик, программист и т.п.).

Разработка осуществляется в соответствии с графиком, в котором указывается этап разработки (код этапа, наименование этапа), дата начала и дата окончания выполнения данного этапа. После выполнения того или иного этапа проставляются фактические сроки выполнения данного этапа разработки.

Вариант № 31

БД программного обеспечения (ПО) дисплейных классов

На кафедре ведется учет установленного в дисплейных классах ПО. О каждом дисплейном классе хранится следующая информация: № дисплейного класса, сотрудник, отвечающий за данный класс (код сотрудника, ФИО сотрудника), а также перечень компьютеров (уникальный инвентарный номер) и их характеристики (код характеристики, название характеристики, значение).

При установке ПО на конкретный компьютер, фиксируется дата установки и сведения о ПО (название, № версии, фирма разработчик (код фирмы разработчика, название фирмы), срок действия лицензии). Одно и то же ПО может быть установлено на нескольких компьютерах.

Вариант № 32

БД выдачи лекарственных препаратов льготникам

В поликлиниках города имеется утвержденный список препаратов (уникальный шифр, название) предназначенных для пациентов, имеющих право на льготы. Льготные рецепты выписываются пациентам, имеющим на это право. О каждом таком пациенте имеется следующая информация: № страхового полиса, ФИО, дата рождения, основание для получения льготных рецептов (код основания, наименование основания), основное заболевание (код МКБ (международный классификатор болезней), наименование заболевания), а также перечень сопутствующих заболеваний. Каждому заболеванию соответствует свой список препаратов.

Выдача льготных препаратов пациентам осуществляется на основании рецептов, в которых указывается № рецепта (уникальный), дата выписки, ФИО врача, ФИО пациента, название препарата, кол-во, а также фиксируется для лечения каких заболеваний назначен данный препарат пациенту, которому выписан рецепт. Один и тот же препарат может быть назначен для лечения нескольких заболеваний. На каждый препарат выписывается свой рецепт. При выдаче препаратов фиксируется дата выдачи, № рецепта, наименование препарата, количество.

Методические материалы

**РАБОТА С ЦЕНТРАЛИЗОВАННЫМИ И РАСПРЕДЕЛЕННЫМИ БАЗАМИ
ДАНЫХ**

Методические указания

Составители *Либерзон Ольга Карловна, Чигарина Елена Ивановна*

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЕВА»
(Самарский университет)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Изд-во Самарского университета.
443086 Самара, Московское шоссе, 34.