

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (СГАУ)

ОСНОВЫ СИСТЕМ
АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Электронные методические указания
к лабораторным работам

Самара
2013

УДК 004.9 (075)
О 753

Автор-составитель: **Нестеренко Елена Сергеевна**

Основы систем автоматизированного проектирования [Электронный ресурс]: электрон. метод. указания к лаб. работам / Минобрнауки России, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т); авт.-сост. Е.С. Нестеренко. – Электрон. текстовые и граф. дан. (0,38 Мбайт). - Самара, 2013.

В лабораторных работах студенты изучают возможности баз данных: СУБД Visual FoxPro, программы Query Analyzer СУБД SQL Server, MS Access.

Указания предназначены для студентов инженерно-технологического факультета, обучающихся по направлению подготовки 150400.62 «Металлургия», изучающих дисциплину «Основы САПР и численные методы» в 5 семестре; 150700.62 «Машиностроение», изучающих дисциплину «Основы САПР и численные методы» в 5 семестре.

Подготовлено на кафедре обработки металлов давлением.

© Самарский государственный
аэрокосмический университет, 2013

СОДЕРЖАНИЕ

Лабораторная работа №1	4
Лабораторная работа № 2	12
Лабораторная работа № 3	22
Лабораторная работа № 4	29
Лабораторная работа № 5	35
Лабораторная работа № 6	40
Лабораторная работа № 7	47
Лабораторная работа № 8	51

Сведения о базах данных

1 Теоретические сведения

База данных – совокупность данных, представленных определенным образом. Или иначе, набор информации о какой-то определенной области.

Данные в базах данных Access содержатся в таблицах, которые по своей структуре и внешнему виду похожи на электронные таблицы Excel. Как и рабочие листы Excel, они состоят из столбцов и строк. Столбцы содержат поля, а в строках, в свою очередь, находятся записи. Поля представляют собой отдельный набор данных, определяющий, например, фамилию заказчика, номер модели, код товара и т.д. Запись представляет собой совокупность полей, которые описывают тот или иной объект, независимо от того, является ли объект автомобилем, человеком или, например, организацией.

Запуск MS Access

Для запуска нажмите кнопку **Пуск**, выполните команду **Программы→Microsoft Access**. При первом запуске Microsoft Access предлагается три варианта начала работы – **Новая база данных, Мастера, страницы и проекты баз данных**, а также **Открыть базу данных**.

Создание новой базы данных

Если хотите начать работу с Access с создания новой базы данных, выберите переключатель **Новая база данных** и нажмите кнопку **ОК**. Введите имя создаваемой базы данных в поле **Имя файла** диалогового окна **Файл новой базы данных** и щелкните на кнопке **Создать** (Access предлагает в

качестве имени базы данных аббревиатуру db1, db2 и т.д., но лучше присвоить файлу более содержательное имя).

Создание базы данных в режиме конструктора

Чтобы создать таблицу в режиме конструктора, щелкните на кнопке **Конструктор** на панели окна базы данных, или дважды щелкните «мышью» на ярлыке **Создание таблицы в режиме конструктора**. Таблица, представленная на рисунке 1, состоит из трех столбцов: **Имя поля**, **Тип данных**, **Описание**.

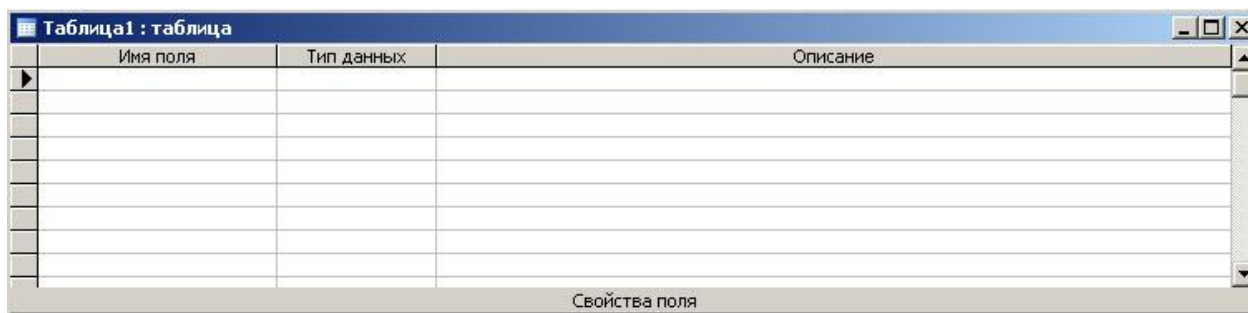


Рисунок 1 – Таблица в режиме конструктора

Имя поля может содержать до 64 символов, но в качестве имени рекомендуется использовать аббревиатуры или краткие названия, например, «Фамилия», «Код товара», «Номер факса» и т.д. Тип данных используется для определения типа данных, хранимых в этом поле. Описание поля не является обязательным параметром таблицы. Если имя выбрано должным образом, дополнительное описание просто не потребуется.

При выборе типа данных, содержащихся в полях таблицы, на вкладке **Общие** в нижней части окна отображаются дополнительные параметры полей, определяющие условия ввода данных. Параметры полей могут несколько отличаться друг от друга в зависимости от типа вводимых данных. Например, для таких типов данных, как **Числовой** или **Денежный**, можно определить число десятичных знаков, а также указать формат вывода

значений данного поля. Подобные параметры можно определить и для типов данных *Дата/Время* или *Счетчик*.

Таблица 1 – Основные типы данных в MS Access

Текстовый	Текстовые поля могут содержать буквы, цифры и специальные символы. Максимальная ширина поля составляет 255 символов
Поле МЕМО	Используется для хранения тех же типов данных, что и текстовые поля, но может содержать до 65535 символов
Числовой	Используется для хранения числовых типов данных
Дата/Время	Используется для хранения дат и времени
Денежный	Используется для хранения данных денежного типа
Счетчик	Используется для хранения уникальных данных, назначаемых автоматически при добавлении каждой новой записи в таблицу
Логический	Используется для хранения данных, которые могут принимать одно из двух возможных значений: Да/Нет, Истина/Ложь и т.п.

2 Запросы

Запросы используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным.

Для поиска и вывода данных, удовлетворяющих заданным условиям, включая данные из нескольких таблиц, необходимо создать запрос. Запрос

также может обновлять или удалять несколько записей одновременно и выполнять стандартные или пользовательские вычисления с данными.

В запросе может выполняться доступ к разным таблицам для отображения кода заказа, названия компании, города и даты исполнения для заказчиков из определенного города, сделавших заказы, которые следует выполнить в одном месяце.

2.1 Создание простых запросов

Запросы можно создавать, используя *мастер создания запросов* и используя *конструктор*.

Наиболее просто создается запрос при помощи *Мастера запросов*. Чтобы создать простой запрос с помощью *Мастера запросов*, необходимо:

1. В окне базы данных на панели объектов выбрать ярлык *Запросы*.
2. В списке запросов дважды щелкнуть левой кнопкой «мыши» на ярлыке *Создание запроса с помощью мастера* или нажать на кнопку *Создать* в окне базы данных и в появившемся диалоговом окне *Новый запрос* выбрать *Простой запрос* и нажать на кнопку *ОК*.

3. В появившемся окне *Создание простых запросов* в поле со списком *Таблицы и запросы* выбрать таблицу или запрос, которые будут служить источником данных для создаваемого запроса.

С помощью стрелок вправо и влево переместить из списка *Доступные поля* в список *Выбранные поля* те поля, которые необходимы в конструируемом запросе. При этом порядок полей в запросе будет соответствовать порядку полей в списке *Выбранные поля*. Если нужно включить в запрос все поля, можно воспользоваться кнопкой с двумя стрелками вправо.

4. Нажать кнопку Далее (Next)

5. Следующее диалоговое окно будет последним. В нем нужно ввести имя создаваемого запроса и выбрать дальнейшие действия. При

необходимости можно установить флажок *Вывести справку по работе с запросом* для вывода справочной информации по работе с запросами.

6. Нажать на кнопку *Готово*.

2.2 Создание и изменение запроса в режиме Конструктора

Для изменения уже существующих запросов и для создания новых запросов используется *Конструктор запросов*. Для того чтобы открыть запрос в режиме *Конструктора*, выделите в списке один из существующих запросов и нажмите кнопку *Конструктор* на панели инструментов окна **База данных**.

Появляется окно *Конструктора запросов*. В верхней части окна отображается таблица (или несколько таблиц, если запрос многотабличный) в том виде, в каком таблицы отображаются в окне *Схема данных*. Таблицы - источники данных для запроса, мы будем называть *базовыми* таблицами запроса. В нижней части окна находится *бланк запроса* - таблица, ячейки которой используются для определения запроса. В бланке отображаются все столбцы, включенные в результирующее множество запроса.

Для того чтобы просмотреть полностью бланк запроса и все исходные таблицы, используют линейки прокрутки.

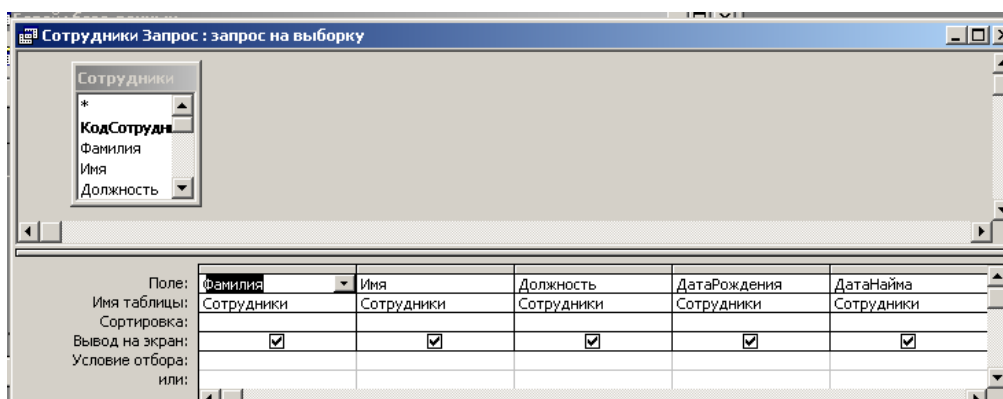


Рисунок 2 – Вид запроса в режиме Конструктора

Рассмотрим, как создать новый запрос с помощью *Конструктора запросов*. Для этого необходимо:

1. В окне базы данных на панели объектов выбрать ярлык *Запросы*.
2. В списке запросов выбрать ярлык *Создание запроса в режиме конструктора* или нажать кнопку *Создать*, в появившемся окне *Новый запрос* выбрать *Конструктор* и нажать на кнопку *ОК*.

3. В окне *Добавление таблицы* выбрать одну или несколько таблиц или запросов для построения нового запроса и нажать кнопку *Добавить*. Для удобства выбора таблиц и запросов в окне существуют следующие вкладки: *Таблицы*, на которой отображается список таблиц; *Запросы*, на которой отображается список запросов; *Таблицы и запросы*, на которой отображается список таблиц и запросов вместе.

4. После добавления всех необходимых таблиц нажать кнопку *Заккрыть* в окне *Добавление таблицы*. Все выбранные таблицы оказываются помещенными на верхней панели окна *Конструктора запросов*. Если таблицы связаны между собой, т. е. связи присутствуют явно на схеме данных, то эти связи также отображаются. Если связи на схеме данных не установлены, то *Конструктор запросов* автоматически устанавливает связи между таблицами, если они содержат поля, которые имеют одинаковые имена и согласованные типы.

5. Затем нужно указать, какие поля из таблиц будут отображаться в запросе. Включать в запрос можно поля из любой таблицы. Способов включения полей в запрос существует несколько. Выделите нужное поле в таблице-источнике (можно выделить несколько полей, пользуясь клавишами <Shift> и <Ctrl>). Если требуется включить в запрос все поля базовой таблицы, выделите поле, обозначенное звездочкой (*). Дважды щелкните левой кнопкой «мыши» на выделенном поле. При этом в бланке запроса появится столбец, соответствующий выбранному полю. Затем аналогично добавьте другие поля. Столбцы в бланке запроса при этом заполняются слева направо.

Можно подвести указатель «мыши» к выделенному полю (одному из выделенных полей), нажать на левую кнопку «мыши» и перетащить поле (поля) в нужное место бланка запроса. Последний способ позволяет помещать поля в любое место бланка запроса.

И, наконец, вместо перетаскивания полей в бланк запроса из таблицы можно просто использовать раскрывающийся список полей в строке *Поле* бланка запроса.

2.3 Элементы языка запросов

- -В данной записи допускается любая последовательность символов. Например, чтобы вывести информацию обо всех сотрудниках, чьи фамилии начинаются на букву «С», в столбец с полем «Фамилия» в строке условие отбора необходимо указать С*. Access автоматически изменит введенное выражение на like «С*», т.е. содержится С, а далее может быть любая последовательность символов.

- Также можно использовать различные математические операторы, например >, <, <> и т.д.

- Если необходимо вывести информацию, где условием отбора является некоторый временной промежуток, то необходимо использовать конструкцию «between ... and». Например, вывести информацию о работниках , которые приняты с 1 мая 2006 года по 10 мая 2007 года. Подобный запрос будет выглядеть следующим образом: для поля «дата найма» в строку условие отбора ввести выражение between 01.05.2006 and 10.05.2007.

3 Формы

Формы являются типом объектов базы данных, который обычно используется для отображения данных в базе данных. Форму можно также использовать как кнопочную форму, открывающую другие формы или

отчеты базы данных, а также как пользовательское диалоговое окно для ввода данных и выполнения действий, определяемых введенными данными.

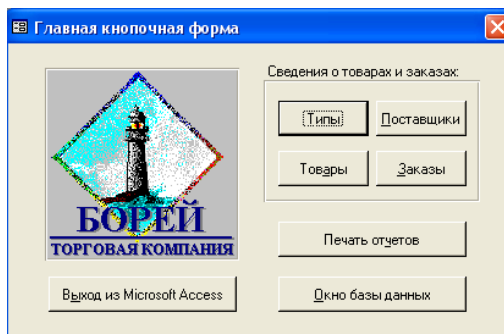


Рисунок 3 –Пример формы

4 Отчеты

Отчет является эффективным средством представления данных в печатном формате. Имея возможность управлять размером и внешним видом всех элементов отчета, пользователь может отобразить сведения желаемым образом.

Для анализа данных или представления их определенным образом в печатном виде, можно создать отчет. Например, напечатать один отчет, группирующий данные и вычисляющий итоговые значения, и еще один отчет с другими данными, отформатированными для печати почтовых наклеек.

5 Макросы

Макрос представляет набор макрокоманд. Макросы могут быть полезны для автоматизации часто выполняемых задач. Например, при нажатии пользователем кнопки можно запустить макрос, который распечатает отчет.

При создании макроса пользователь вводит макрокоманды, которые требуется выполнить.

Лабораторная работа № 2

Изучение возможностей СУБД Visual FoxPro по созданию и ведению баз данных

1. Создать папку для размещения файлов базы данных. Запустить Visual FoxPro. Установить созданную папку в качестве директории по умолчанию, установить нужные форматы даты и времени:

Сист. меню | Tools | Options... =>

форма Options |

(стр. File Locations | выбр. Default Directory, кн. Modify =>

(форма Change File Location | Use Default Directory <- вкл.;

кн. [...] => (форма Select Directory | выбрать папку; кн.

Select);

кн. ОК);

кн. Set As Default),

(стр. Regionals | Data Format <- 'German', Date Separator <- вкл., <-

!!;

Century <- вкл., 24-Hour <- вкл.; Set As Default);

кн. ОК.

2. Создать новый проект, в нем создать новую базу данных:

Сист. меню | File | New... => ф. New | Project <- вкл.; кн. New File =>

(форма Create | Enter Project <- Имя проекта, напр., 'Edu_vfp';

кн. Сохранить) =>

(форма Project Manager | стр. Data | выбр. Databases; кн. New) =>

(форма Create | Enter Database <- имя базы данных, напр., 'Education';

кн. Сохранить) =>

(форма Database Designer <- закрыть);

Project manager | [+]Databases <- раскрыть | [+]Education <- раскрыть.

3. В базе данных создать структуру таблицы Student (справочник студентов) с полями для номера зачетной книжки no_zk, фамилии fam, имени im, отчества ot, кода специальности spes (все строковые), курса kurs (числовое), номера группы gr (строковое), даты рождения datar, даты поступления в вуз datar (дата), биографии biogr (мемо), фотографии foto (объект):

Project Manager | стр. Data | Databases | Education | выбр. Tables; кн. New =>

(форма Create | Enter Table <- 'Student'; кн. Сохранить) =>

(форма Table Designer | стр. Field |

Name <- 'no_zk'; Type <- 'Character'; Width <- 6; на след. строку;

Name <- 'fam'; Type <- 'Character'; Width <- 20; на след. строку;

... и т.д. для всех полей;

кн. ОК) =>

(форма Input data records now? | кн. No)

4. Аналогичным образом создать структуру таблицы Predmet (справочник предметов) с полями для идентификатора предмета no_pr, названия предмета nazv (строковое), количества часов kol_chas (числовое), кода специальности spes (строковое).

5. Аналогичным образом создать структуру таблицы Uspev (успеваемость студента по предмету) с полями для номера зачетной книжки (как в таблице Student), идентификатора предмета (как в таблице Predmet), оценки osenka (символ), даты data (дата).

6. Через Browse-окно наполнить базу данных сведениями о студентах (не менее 4-х), предметах (не менее 3-х) и оценках (не менее 7).

Project Manager | стр. Data | Databases | Education | Tables |

выбр. Student; кн. Browse => (окно содержимого таблицы);

Сист. меню | View | Browse – табличный формат Browse-окна,

Сист. меню | View | Append Mode – разрешение добавления записей;

Ввести значения полей в строки (кроме полей General);

[X] – закрыть окно с сохранением изменений)

– освоить логическое удаление записей:

Project Manager | стр. Data | Databases | Education | Tables |

выбр. Student; кн. Browse => (окно содержимого таблицы);

клик на безымянном левом поле записи =>

поле черное – признак логически удаленной записи;

повторный клик – признак снимается – восстановление записи

Окно Command Window | команда SET DELETED ON – невидимость удаленных

Окно Command Window | команда SET DELETED OFF – видимость удаленных

– освоить физическое удаление логически удаленных записей:

Browse-окно | добавить новую запись; логически удалить (только ее!);

Окно Command Window | команда PACK – сжатие таблицы

Browse-окно | убедиться в отсутствии удаленной записи.

7. Для General-полей опробовать различные варианты внедрения и связывания объектов, а также приемы их модификации.

– внедрение в поле объекта из OLE-сервера:

Browse-окно | клик по пустому General-полю Foto => (Окно General;

Системное меню | Edit | Insert Object =>

(форма Insert Object | (Choose | Create New <- вкл.),

Object Type <- 'Paint Brush Picture'; кн. ОК =>

(Окно Paint | нарисовать "фото"; клик за окном Paint) =>

"фото" в окне General; [X])

метка 'Gen' в поле Foto.

– внедрение в поле объекта из файла:

Создать или скопировать заранее BMP-файл — "фото"

Browse-окно | клик по пустому General-полю Foto => (Окно General;

Системное меню | Edit | Insert Object =>

(форма Insert Object | (Choose | Create from File <- вкл.),

Link <- откл.; кн. Browse =>

(Окно Browse | селектировать файл с "фото"; [X]);

кн. Открыть) =>

"фото" в окне General; [X])

метка 'Gen' в поле Foto.

– связывание поля с объектом из файла:

Создать или скопировать заранее BMP-файл — "фото"

Browse-окно | клик по пустому General-полю Foto => (Окно General;

Системное меню | Edit | Insert Object =>

(форма Insert Object | (Choose | Create from File <- вкл.),

Link <- вкл.; кн. Browse =>

(Окно Browse | выбрать файл с "фото"; [X]);

кн. Открыть) =>

"фото" в окне General; [X])

метка 'Gen' в поле Foto.

– модификация объекта:

Browse-окно | клик по непустому General-полю Foto =>

(Окно General | клик =>

(Окно Paint | изменить "фото"; клик за окном Paint) =>

измененное "фото" в окне General; [X]).

– удаление объекта:

Browse-окно | клик по непустому General-полю Foto => (Окно General;

Системное меню | Edit | Clear =>

исчезло "фото" в окне General; [X]) =>

метка 'gen' в поле Foto.

8. В таблице студентов ввести ограничения для двух-трех полей с сообщениями пользователю, значения по умолчанию, формат поля, заголовок:

Project Manager | стр. Data | Databases | Education | Tables | Student;

кн. Modify =>

(форма Table Designer | стр. Field |

(выбр. поле Kurs;

Field Validation | (Rule <- 'kurs>=1 AND kurs<=6',

Message <- ' "Неверный курс" ', Default Value <- 1),

Display | Caption <- 'Курс'),

выбр. другое поле ... аналогично ... и т.д. для других полей... кн. ОК)

=>

(форма Table Designer: Compare the rules... | кн. Yes)

если сущ. данные не противоречат новым ограничениям, =>)

– ввести ограничение на уровне записи с сообщением пользователю:

Project Manager | стр. Data | Databases | Education | Tables | Student;

кн. Modify =>

(форма Table Designer | стр. Table |

(Record Validation | (Rule <- 'dataP-dataR<=17*365',

Message <- ' "Несоответствие дат рождения и поступления в вуз"

'));)

кн. ОК) => (форма Table Designer: Compare the rules... | кн. Yes)

если сущ. данные не противоречат новым ограничениям, =>)

– убедиться в работоспособности ограничений: новые заголовки полей; значение по умолчанию; сообщения при нарушении ограничений; выяснить, в какой момент проверяются ограничения на уровне поля и на уровне записи.

9. В таблицах студентов и предметов создать первичные индексы по номеру зачетки и идентификатору предмета соответственно, а в таблице успеваемости — соответствующие регулярные индексы:

Project Manager | стр. Data | Databases | Education | Tables | Student;

кн. Modify =>

(форма Table Designer |

(стр. Table | (выбр. поле no_zk; клик в столбце Index =>

(раскр. список <- 'Ascending'));

(стр. Index | клик в Type => (раскр. список <- 'Primary'));

кн. ОК) => (форма Table Designer: Compare the rules... | кн. Yes)
если сущ. записи не содержат дубликатов поля no_zk, =>)

Аналогичным образом для таблиц предметов и успеваемости (в таблице успеваемости не исправлять индекс Regular на Primary);

– убедиться в том, что первичный индекс препятствует внесению записей с дубликатами ключа;

– освоить использование индексов при просмотре через окно Browse:

Панель инструментов | кн. Data Session Window =>

(форма Data Session |

Если в списке Aliases нет 'Uspev', то кн. Open =>

(форма Open | сп. Tables in database <- 'Uspev'; кн. ОК);

выд. 'Uspev'; кн. Properties =>

(форма Work Area Properties |

раскр. сп. Index Order <- 'Uspev: no_zk'; кн. ОК);

выд. 'Uspev'; кн. Browse =>

10. Через окно дизайнера базы данных связать таблицы, сделав таблицы студентов и предметов родительскими, а таблицу успеваемости — дочерней; установить правила ссылочной целостности: каскадное для удаления и изменения; строгое для вставки:

Project Manager | стр. Data | Databases | Education; кн. Modify =>

(форма Database Designer |

(перетащить Student | Indexes | no_zk -> Uspev | Indexes | no_zk)

=>

связь Student ->> Uspev,

(перетащить Predmet | Indexes | no_pr -> Uspev | Indexes | no_pr)

=>

связь Predmet ->> Uspev;

(дв.клик по одной из связей =>

(форма Edit Relationship | кн. Referential Integrity =>

(форма Referential Integrity Builder |

клик Student в таблице связей;

(стр. Rules for Updating | перекл. <-

Cascade),

(стр. Rules for Deleting | перекл. <-

Cascade),

(стр. Rules for Inserting | перекл. <-

Restrict);

клик Student в таблице связей;

(стр. Rules for Updating | перекл. <-

Cascade),

(стр. Rules for Deleting | перекл. <-

Cascade),

(стр. Rules for Inserting | перекл. <-

Restrict); кн. ОК =>

(форма Do you want... | кн. ОК) =>

(форма Предупреждение... | кн. ОК) => ...)))

Если сообщение о необходимости Clean Up Database, то:

Системное меню | Database | Clean Up Database;

– проверить действие правил ссылочной целостности, удаляя, изменяя и вставляя данные.

11. Создать локальный просмотр S_view, отображающий из таблицы Student: номер зач. кн.; фамилию и инициалы; специальность - курс - группу в стандартной форме (например, АСУ-314):

Project Manager | стр. Data | Databases | Education | Local Views | кн. New =>
(форма View Designer |

(форма Add Table or View | Tables in database | дв.клик 'Student';
кн. Close);

(стр. Fields |

(Avaliable fields | дв.клик 'Student.no_zk' =>
перенос в Selected fields);

(Functions and expressions <-

'TRIM(fam)+" "+LEFT(im,1)+"."+LEFT(ot,1)+"." AS

fio';

кн. ADD => перенос в Selected fields);

(Functions and expressions <-

'TRIM(спеc)+"-"+TRIM(STR(kurs))+gr AS группа';

кн. ADD => перенос в Selected fields));

кн. [X]) => (форма Do you want to save changes? | кн.ОК) =>

(форма Save | View Name <- 'S_view'; кн.ОК) =>).

Project Manager | стр.Data | Databases | Education | Local Views |

'S_view'; кн. Browse => содержимое просмотра.

12. Создать локальный просмотр U_S_P_view для вывода сведений об успеваемости студентов из таблицы Uspev с указанием сведений о студенте из локального просмотра S_view и сведений о предмете из таблицы Predmet.

Project Manager | стр. Data | Databases | Education | Local Views | кн. New =>
(форма View Designer |

(форма Add Table or View | Tables in database |
дв.клик 'Uspev'; дв.клик 'Predmet';
(Select | клик Views); дв.клик 'S_view'; кн. Close);
(стр. Fields | кн. ADD ALL =>
перенос всех в Selected fields);
кн. [X]) => (форма Do you want to save changes? | кн.ОК) =>
(форма Save | View Name <- 'U_S_P_view'; кн.ОК) =>).

Project Manager | стр.Data | Databases | Education | Local Views |
'U_S_P_view'; кн. Browse => содержимое просмотра.

**Изучение возможностей СУБД Visual FoxPro по созданию
простых отчетных и экранных форм**

1. Убедиться в наличии папки с базой данных, созданной при выполнении предыдущей лабораторной работы. Запустить Visual FoxPro, настроить его на нужную папку, открыть проект и проверить работоспособность базы данных.

5.1 Отчетные формы

2. Построить форму 1-табличного отчета с помощью построителя Quick Report для таблицы студентов:

Project Manager | стр. Data | Databases | Education | Tables | выбр. Student;

Панель инструментов | кн. AutoReport Wizard =>

(ф. Report Designer с формой отчета |

Сист. меню | File | Save As =>

ф. SaveAs | SaveReport <- Student_Quik_Report | кн. Сохранить;

кн. [X] =>)

Project Manager | стр. Documents | выдел. Reports; кн. Add =>

(ф. Open | Select Report <- Student_Quik_Report; кн. ОК =>)

Project Manager | стр. Documents | [+] Reports раскр. | выдел. Student_Quik_Report; кн. Preview =>

(отчет на экране | изучить; кн. [X] =>)

3. Построить форму 1-табличного отчета с помощью мастера Report Wizard для таблицы студентов:

Project Manager | стр. Documents | выдел. Reports; кн. New =>

(ф. New Report | кн. Report Wizard => ф. Wizard Selection | выдел. Report Wizard; кн. ОК =>

(ф. Report Wizard |

Step 1 | Databases and tables <- Education, Student;

(кн. [>>] => Selected fields <- все поля); кн. Next =>

Step 2 | кн. Next => Step 3 | кн. Next => Step 4 | кн. Next => Step 5 |
кн. Finish =>

(ф. Save As | Save report as <- Student; кн. Сохранить) => (...) =>
отчетная форма готова.

Project Manager | стр. Documents | Reports | выбр. Student; кн. Preview =>
отчет на экране;

– разобраться в вариантах функционирования и оформления отчетной
формы, создаваемой мастером:

Повторить несколько раз, варьируя Style на Step 2;

Number of columns, Field layout, Orientation на Step 3;

Selected fields, Ascending/Descending на Step 4.

4. Аналогичным образом построить и проверить отчетную форму для
таблицы предметов, выбрав для нее другие параметры функционирования и
оформления.

5. Построить форму 2-табличного отчета ("один ко многим") с помощью
мастера One to Many Report Wizard для таблицы студентов в качестве
родительской и таблицы успеваемости в качестве дочерней:

Project Manager | стр. Documents | выбр. Reports; кн. New =>

(ф. New Report | кн. Report Wizard) =>

(ф. Wizard Selection | выбр. One-to-Many Report Wizard; кн. ОК) =>

(ф. One-to-Many Report Wizard |

Step 1 | Databases and tables <- Education, Student;

(кн. [>>] => Selected fields <- все поля); кн. Next =>

Step 2 | Databases and tables <- Education, Uspev;

(кн. [>>] => Selected fields <- все поля); кн. Next =>

Step 3 | кн. Next => Step 4 | кн. Next => Step 5 | кн. Next => Step 6 |

кн. Finish =>

(ф. Save As | Save report as ← *Student_Uspev*; кн. Сохранить) => (...) =>
форма готова.

Project Manager | стр. Documents | Reports | выбр. *Student_Uspev*; кн. Preview
=>

отчет на экране.

– разобраться в вариантах функционирования и оформления отчетной формы, создаваемой мастером.

6. Аналогичным образом построить и проверить 2-табличную отчетную форму для таблиц предметов и успеваемости, выбрав для нее другие параметры функционирования и оформления.

7. Освоить приемы простейшей модификации созданных форм отчетов с помощью конструктора отчетов Report Designer:

Project Manager | стр. Documents | Reports | выдел. *Student*; кн. Modify =>

(ф. Report Designer с формой отчета |

– изменение расположения объектов:

(кл. на объекте => выделение); перетащить на новое место;

(кл. на объекте; (удерж. Shift + кл. на др. объекте) => добавл. к

выделенным;

перетащить неск. объектов на новое место;

– изменение размера объектов:

(кл. на объекте => выделение); перетащить черн. квадратик на
новое место;

– корректировка надписей:

Сист. меню | View | Report Controls Toolbar ← отметка =>

панель. Report Controls | кн. A; клик на надписи =>

(режим редактирования — *исправить надпись*;

клик за пределами надписи =>)

– изменение шрифта надписи:

(кл. на надписи => выделение); Сист. меню | Format | Font =>

(ф. Выбор шрифта | установить нужные атрибуты шрифта;
кн. ОК =>)

– изменение расстояния между строками:

Полосу Details перетащить вниз или вверх;

– отдельная титульная страница :

Сист. меню | Report | Title/Summary => (ф. Title/Summary | уст.
New Page; кн. ОК =>)

– выйти из Report Designer и проверить исправления:

Report Designer | кн. [X] => ф. Do you want to save changes...? | кн.
Yes)

Project Manager | стр. Documents | Reports | выбр. Student; кн. Preview =>
отчет Student на экране

Э к р а н н ы е ф о р м ы

8. Освоить быстрое создание однотабличных форм автоформатированием:

Project Manager | стр. Data | Databases | Education | Tables | выбр. Student;

Панель инструментов | кн. AutoForm Wizard => (...) => форма готова;

изучить назначение и действие кнопок полученной формы.

9. Освоить создание однотабличной формы с помощью мастера:

– запустить мастер форм (Form Wizard) и с его помощью построить и
сохранить простую форму для таблицы студентов. Запустить построенную
форму на выполнение, убедиться в ее работоспособности и правильном
выполнении функций:

Project Manager | стр. Documents | выбр. Forms; кн. New =>

(ф. New Form | кн. Form Wizard) =>

(ф. Wizard Selection | выбр. Form Wizard; кн. ОК) =>

(ф. Form Wizard |

Step 1 | Databases and tables <- Education, Student;

(кн. [>>] => Selected fields <- все поля); кн. Next =>

Step 2 | кн. Next => Step 3 | кн. Next => Step 4 | кн. Finish =>

(ф. Save As | Save form as <- Student; кн. Сохранить) => (...) => форма

готова.

Project Manager | стр. Documents | раскр. [+] Forms | выбр. Student; кн. Run =>

форма на экране.

– разобраться в вариантах функционирования и оформления экранной формы, создаваемой мастером:

Повторить несколько раз, варьируя Style, Button type на Step 2;

Selected fields, Ascending/Descending на Step 3.

10. Аналогичным образом построить и проверить форму для таблицы предметов, выбрав для нее другие параметры функционирования и оформления.

11. С помощью мастера форм One to Many Form Wizard построить 2-табличную форму ("один ко многим") для таблицы студентов в качестве родительской и таблицы успеваемости в качестве дочерней. Запустить построенную форму на выполнение, убедиться в ее работоспособности и правильном выполнении функций:

Project Manager | стр. Documents | выбр. Forms; кн. New =>

(ф. New Form | кн. Form Wizard) =>

(ф. Wizard Selection | выбр. One-to-Many Form Wizard; кн. ОК) =>

(ф. One-to-Many Form Wizard |

Step 1 | Databases and tables <- Education, Student;

(кн. [>>] => Selected fields <- все поля); кн. Next =>

Step 2 | Databases and tables <- Education, Uspev;

(кн. [>>] => Selected fields <- все поля); кн. Next =>

Step 3 | кн. Next => Step 4 | кн. Next => Step 5 | кн. Next => Step 6 |

кн. Finish =>

(ф. Save As | Save form as <- *Student_Uspev*; кн. Сохранить) => (...) =>
форма готова.

Project Manager | стр. Documents | Forms | выбр. *Student_Uspev*; кн. Run =>
форма на экране.

– разобраться в вариантах функционирования и оформления экранной формы, создаваемой мастером.

12. Аналогичным образом построить и проверить 2-табличную форму для таблиц предметов и успеваемости, выбрав для нее другие параметры функционирования и оформления.

13. Освоить приемы простейшей модификации созданных форм с помощью конструктора форм (Form Designer):

Project Manager | стр. Documents | Forms | выбр. *Student*; кн. Modify =>
(ф. Form Designer с загруженной формой *Student* |

– изменение расположения объектов:

(кл. на объекте => выделение); перетащить на новое место;

(кл. на объекте; (удерж. Shift + кл. на др. объекте) => добавл. к

выделенным;

перетащить неск. объектов на новое место;

– изменение размера объектов:

(кл. на объекте => выделение); перетащить черн. квадратик на новое место;

– корректировка надписей:

мышь на объект *Label*; прав. кн. мыши => всплыв меню |
Properties =>

ф. Properties | выдел. Caption; поле значения ← *новая надпись*; Enter =>

надпись изменена;

Form Designer | кн. [X] => ф. Do you want to save changes...? | кн. Yes)

Project Manager | стр. Documents | Forms | выбр. Student; кн. Run =>
форма Student на экране

Взаимодействие отчетных и экранных форм

14. Разместить на экранной форме кнопку управления для вывода отчета:

Project Manager | стр. Documents | Forms | выбр. Student; кн. Modify =>

(ф. Form Designer с загруженной формой Student |

сист. меню | View | check Form Controls Toolbar => панель инстр.

Form Controls |

кн. Command Button; кл. на форме => появилась новая кнопка;

мышь на кнопку; прав. кн. мыши => всплыв меню | Properties =>

ф. Properties | выдел. Caption; поле значения ← *Отчет*;

Enter;

дв. кл. на кнопке =>

(окно Command.Click ← *REPORT FORM student PREVIEW*;

кн. [X])

Form Designer | кн. [X] => ф. Do you want to save changes...? | кн. Yes)

Project Manager | стр. Documents | Forms | выбр. Student; кн. Run =>

форма Student | кн. Отчет => (отчет Student на экране | кн. [X] =>
возврат в форму)

Лабораторная работа № 4

Изучение возможностей СУБД Access по созданию и ведению баз данных

1. Создать папку для размещения файла базы данных. Запустить Microsoft Access. Создать новую базу данных:

Пуск | Программы | Microsoft Access | ф. Microsoft Access | уст. Новая база данных | кн. ОК ►

(ф. Файл новой базы данных | Имя файла ← *Education*; кн. Создать) ►

ф. Education: база данных |

2. В базе данных с помощью конструктора таблиц создать структуру таблицы Student (справочник студентов):

ф. Education: база данных | уст. Таблицы; дв.кл. Создание таблицы в режиме конструктора ►

ф. конструктора таблиц |

– задать поле идентификатора студента (счетчик, первичный ключ):

Имя поля ← *stud_ID*; Тип данных ← Счетчик; (панель инстр. | кн. Ключевое поле); дв. Enter;

– задать поле номера зачетной книжки (уникальное, 6 цифр):

Имя поля ← *no_zk*; Тип данных ← Текстовый; (Свойства поля | Общие | Размер поля ← 6; Индексированное поле ← Да (Совпадения не допускаются);

Обязательное поле ← Да; Маска ввода ← 000000); кл. на след. строке;

– задать поля фамилии, имени, отчества (строковые, уникальная совокупность):

Имя поля ← *fam*; Тип данных ← Текстовый; (Свойства поля | Общие |

Размер поля ← 20; Обязательное поле ← Да); кл. на след. строке; аналогично — для *im* и *ot*;

Панель инстр. | кн. Индексы ► (ф. Индексы | кл. на пустую строку | Индекс ← *fio*;

Имя поля ← *fan*; кл. на след. строке | Имя поля ← *im*; кл. на след. строке |

Имя поля ← *ot*; кн. [X] ►); кл. на след. строке;

– задать поле кода специальности (до 7 русских букв, подстановка из списка):

Имя поля ← *spes*; Тип данных ← Текстовый; (Свойства поля | Общие | Размер поля ← 7; Условие на значение ← *Between "A" And "яаяаяяя"*;

Обязательное поле ← Да; Тип данных ← Мастер подстановок...



(ф. Создание подстановки | уст. Будет введен фиксированный набор значений;

кн. Далее ► след. стр. | Число столбцов ← 2;

сжать 1-й, растянуть 2-й столбец; кн. Далее ► след. стр. |

Доступные поля ← Столбец1; кн. Далее ► след. стр. | кн.

Готово ►);

(Свойства поля | Подстановка | Источник строк ←

АСОИУ; Автоматизированные системы...; МИЭ;

Матметоды в экон...

и т.д. для др. специальностей); кл. на след. строке;

– задать поле курса (одна цифра от 1 до 6):

Имя поля ← *kurs*; Тип данных ← Текстовый; (Свойства поля | Общие |

Размер поля ← 1; Обязательное поле ← Да;

Условие на значение ← *Between "1" And "6"*); кл. на след. строке;

– задать поле номера группы (две цифры):

Имя поля ← *gr*; Тип данных ← Текстовый; (Свойства поля | Общие |

Размер поля ← 2; Обязательное поле ← Да; Маска ввода ← 00);
кл. на след. строке;

– задать поле даты рождения:

Имя поля ← *data_r*; Тип данных ← Дата/время; (Свойства поля |
Общие |

Формат поля ← Краткий формат даты); кл. на след. строке;

– задать поле биографии (мемо):

Имя поля ← *biogr*; Тип данных ← Поле МЕМО; (Свойства поля |
Общие |

Формат поля ← @; "Неизвестна"); кл. на след. строке;

– задать поле адреса домашней страницы (гиперссылка):

Имя поля ← *home_page*; Тип данных ← Гиперссылка; кл. на след.
строке;

– задать поле фотографии (объект):

Имя поля ← *foto*; Тип данных ← Поле объекта OLE; кл. на след.
строке;

– сохранить структуру таблицы:

кн. [X]) ► (ф. Сохранить ...? | кн. Да) ► (ф. Сохранение | Имя таблицы
← *Student*; кн. ОК ►)

3. Аналогичным образом создать структуру таблицы *Predmet* (справочник предметов) с полями для идентификатора предмета (счетчик), названия предмета (строковое), количества часов (числовое), кода специальности (строковое).

4. Аналогичным образом создать структуру таблицы *Uspev* (успеваемость студента по предмету) с полями идентификатора студента (целое, как в таблице *Student*, но не счетчик), идентификатора предмета (целое, как в

таблице Predmet, но не счетчик), оценки (символ), даты (дата). Ввести составной первичный ключ (поля идентификаторов студента и предмета):

ф. конструктора таблицы Uspev | кл. на поле stud_ID; Ctrl + кл. на поле predm_ID ►

выделены оба поля; (панель INSTR. | кн. Ключевое поле) ►
составной ключ готов;

– задать таблицы студентов и предметов в качестве таблиц подстановки для идентификаторов студента и предмета в таблице успеваемости:

ф. конструктора таблицы Uspev | кл. на поле stud_ID; Тип данных | Мастер подстановок... ►

(ф. Создание подстановки | уст. ...значения из таблицы или запроса; кн. Далее ► след. стр. |

кл. Student; кн. Далее ► след. стр. | Выбранные поля <<— поля stud_ID, no_zk, fam,

im, ot, spec, kurs,gr; кн. Далее ► след. стр. | отрегулировать ширину столбцов;

кн. Далее ► след. стр. | кн. Готово) ► ф. ...сохранить таблицу? | кн. Да)

5. Связать таблицы, сделав таблицы студентов и предметов родительскими, а таблицу успеваемости — дочерней; установить правила ссылочной целостности: каскадное для удаления и изменения:

ф. Education: база данных; панель инструментов | кн. Схема данных ► (окно Схема данных |

панель инструментов | кн. Отобразить таблицу ► (ф. Добавление таблицы | Таблицы |

дв.кл. Student, дв.кл. Predmet, дв.кл. Uspev; кн. Закрыть) ► таблицы на схеме;

(Student | student_ID перетащить на Uspev | student_ID) ►

(ф. Изменение связей | уст. Обеспечение целостности; уст. Каскадное обновление;

уст. Каскадное удаление; кн. Создать) ► связь на схеме;

(Predmet | predmet_ID перетащить на Uspev | predmet_ID) ►

(ф. Изменение связей | уст. Обеспечение целостности; уст. Каскадное обновление;

уст. Каскадное удаление; кн. Создать) ► связь на схеме;

кн. [X] ► (ф. Сохранить изменения макета...? кн. Да) ►)

6. Наполнить базу данных сведениями о студентах (не менее 4-х), предметах (не менее 3-х) и оценках (не менее 7). Освоить доступ к дочерним записям из родительской таблицы. Проверить работоспособность полей и таблиц подстановок. Освоить приемы изменения и удаления полей и записей. Для OLE-полей опробовать различные варианты внедрения и связывания объектов, а также приемы их модификации. Проверить работоспособность ограничений значений полей, уникальности и др., предусмотренные при задании структуры базы данных. Проверить работоспособность ссылочной целостности, удаляя, изменяя и вставляя данные.

7. Создать запрос для вывода кратких сведений о студентах (идентификатор, номер зачетки, фамилия и инициалы, идентификатор группы):

ф. Education: база данных | уст. Запросы; дв.кл. Создание запроса в режиме конструктора ►

(ф. конструктора запросов (ф. Добавление таблицы | Таблицы | дв. кл. Student; кн. Закрывать) |

(1-я кол | Поле ← stud_ID); (2-я кол | Поле ← no_zk); (3-я кол | Поле | Shift + F2 ►

(ф. Область ввода ← *FIO:Trim(fam)&'&Left(Trim(im);1)&'&Left(Trim(im);1)&'&' ; кн. ОК))*

(4-я кол | Поле | Shift + F2 ► (ф. Область ввода ← *Gruppa:Trim(spec)&'&kurs&gr ; кн. ОК))*

кн. [X] ► (ф. Сохранить изменения макета...? кн. Да) ►

(ф. Сохранение | Имя запроса ← *Student_brief*;))

ф. Education: база данных | уст. Запросы; дв.кл. Student_brief ► результаты запроса на экране.

8. Аналогичным образом создать запрос для вывода сведений об успеваемости студентов из таблицы *Uspev* с указанием сведений о студенте из запроса *Student_brief* и сведений о предмете из таблицы *Predmet*.

Изучение возможностей программы Enterprise Manager СУБД SQL Server

по созданию и ведению баз данных

1. Запустить Microsoft SQL Server:

Пуск | Программы | Microsoft SQL Server | Service Manager ►

(ф. SQL Server Service Manager | кн. Start/Continue ► ... MS SQL Server – Running; кн. [X] ►)

2. Запустить Enterprise Manager. Создать новую базу данных на Microsoft SQL Server:

Пуск | Программы | Microsoft SQL Server | Enterprise Manager ►

ф. SQL Server Enterprise Manager | ф. Console Root | (панель | уст. кн. Show/Hide Console Tree);

раскр. [+] Microsoft SQL Servers | раскр. [+] SQL Server Group |

раскр. [+] имя сервера (Windows 95/98);

(прав.кн.мыши на (Windows 95/98) | Edit SQL Server Registration Properties ►

(ф. Registered SQL Server Properties | снять Show system databases..; кн. ОК))

раскр. [+] Databases; прав.кн.мыши на Databases | New Database ►

(ф. Database Properties | Name ← *Education*; кн. ОК)

раскр. [+] Education

2. В базе данных создать структуру таблицы Student (справочник студентов):

ф. SQL Server Enterprise Manager | ф. Console Root | Databases | Education |

(прав.кн.мыши на Tables | New Table ►

(ф. New Table in 'Edu' | (ф. Choose name | Enter name ← *Student*;

кн. ОК);

– задать поле идентификатора студента (счетчик, первичный ключ):

Column Name ← *stud_ID*; Datatype ← int; снять Allow Null; уст. Identity;
(Панель | кн. Set primary key ► метка ключа в начале строки);

– задать поле номера зачетной книжки (уникальное, 6 цифр):

Column Name ← *no_zk*; Datatype ← char; Length ← 6; снять Allow Null;
(Панель | кн. Table and Index Properties) ►

(ф. Properties | Indexes/Keys | кн. New; Column Name ←
no_zk;

Index Name ← *Unique_no_zk*; уст. Create Unique; кн.
Закрывать);

– задать поля фамилии, имени, отчества (строковые, уникальная совокупность):

Column Name ← *fam*; Datatype ← varchar; Length ← 20; снять Allow
Null;

аналогично для *im* и *ot*;

(Панель | кн. Table and Index Properties) ►

(ф. Properties | Indexes/Keys | кн. New;

Column Name ← *fam*; след. Column Name ← *im*; след.
Column Name ← *ot*;

Index Name ← *Unique_fio*; уст. Create Unique; кн. Закрывать);

– задать поле кода специальности (до 7 русских букв):

Column Name ← *spec*; Datatype ← varchar; Length ← 7; снять Allow
Null;

(Панель | кн. Table and Index Properties) ►

(ф. Properties | Tables | кн. New;

Constraint Expression ← (*spec* >= 'А' AND *spec* <= 'я');

Constraint Name ← *Russian_spec*; кн. Закрывать);

– задать поле курса (одна цифра от 1 до 6):

Column Name ← *kurs*; Datatype ← char; Length ← 1; снять Allow Null;

(Панель | кн. Table and Index Properties) ►

(ф. Properties | Tables | кн. New;

Constraint Expression ← (*kurs* ≥ 1 AND *kurs* ≤ 6);

Constraint Name ← *Kurs_from_1_to_6*; кн. Закрыть);

– задать поле номера группы (две цифры):

Column Name ← *gr*; Datatype ← char; Length ← 2; снять Allow Null;

(Панель | кн. Table and Index Properties) ►

(ф. Properties | Tables | кн. New;

Constraint Expression ← (*gr* ≥ '01' AND *gr* ≤ '99');

Constraint Name ← *Gr_from_01_to_99*; кн. Закрыть);

– задать поле даты рождения:

Column Name ← *data_r*; Datatype ← datetime;

– задать поле биографии (мемо):

Column Name ← *biogr*; Datatype ← text;

– задать поле фотографии (объект):

Column Name ← *foto*; Datatype ← image; кл. на след. строке;

– сохранить структуру таблицы:

кн. [X]) ► (ф. Do you want to save ...? | кн. Да).

3. Аналогичным образом создать структуру таблицы Predmet (справочник предметов) с полями для идентификатора предмета (счетчик), названия предмета (строковое), количества часов (числовое), кода специальности (строковое).

4. Аналогичным образом создать структуру таблицы Uspev (успеваемость студента по предмету) с полями идентификатора студента (целое, как в таблице Student, но не счетчик), идентификатора предмета (целое, как в

таблице Predmet, но не счетчик), оценки (символ), даты (дата). Ввести составной первичный ключ (поля идентификаторов студента и предмета):

ф. конструктора таблицы Uspev | кл. на поле stud_ID; Shift + кл. на поле predm_ID ►

выделены оба поля; (панель инстр. | кн. Ключевое поле) ►
составной ключ готов;

5. Связать таблицы, сделав таблицы студентов и предметов родительскими, а таблицу успеваемости — дочерней; установить правила ссылочной целостности:

ф. SQL Server Enterprise Manager | ф. Console Root | Databases | Education |

(прав.кн.мыши на Diagrams | New Database Diagram ►

(ф. New Diagram in 'Edu'

(ф. CreateDatabase Diagram Wizard | кн. Далее ► след. стр. |

дв.клик Student, дв.клик Uspev, дв.клик Predmet; кн.

Далее ► след. стр. |

кн. Готово ► (ф. The tables have been added ... | кн.

OK ►) ►)

(Student | ключ) перетащить на (Uspev) ► (ф. Create Relationships | OK);

(Predmet | ключ) перетащить на (Uspev) ► (ф. Create Relationships | OK);

кн. [X] ► (ф. Do you want to save changes...? | кн. Да) ►

(ф. Save As | Save this diagram as ← *Edu_diagram*; OK)

►

(ф. Save | кн. Yes)

6. Наполнить базу данных сведениями о студентах (не менее 4-х), предметах (не менее 3-х) и оценках (не менее 7). Освоить приемы изменения и удаления полей и записей. Проверить работоспособность ограничений значений полей, уникальности и др., предусмотренные при задании структуры базы данных.

Проверить работоспособность ссылочной целостности, удаляя, изменяя и вставляя данные.

7. Создать просмотр для вывода кратких сведений о студентах (идентификатор, номер зачетки, фамилия и инициалы, идентификатор группы):

ф. SQL Server Enterprise Manager | ф. Console Root | Databases | Education |

(прав.кн.мыши на Views | New View ►

(ф. New View in 'Edu' | панель инстр. | кн. Add Tables ►

(ф. Add Table | дв. кл. Student; кн. Close) |

(1-я стр. | Column ← stud_ID); (2-я стр. | Column ← no_zk);

(3-я стр. | Column ← *RTrim(fam)+'*

+Substring(im;1;1)+'.'+*Substring(im;1;1)+'.'* ; Alias ← fio);

(4-я стр. | Column ← *RTrim(spec)+'-'+kurs+gr* ; Alias ← группа)

кн. [X] ► (ф. Save changes...? | кн. Да) ► (ф. Save As | Save this view as
← *Student_brief*; ОК))

ф. SQL Server Enterprise Manager | ф. Console Root | Databases | Education |
Views |

прав.кл. Student_brief | Open view | Return all rows ► результаты на
экране.

8. Аналогичным образом создать просмотр для вывода сведений об успеваемости студентов из таблицы Uspev с указанием сведений о студенте из запроса Student_brief и сведений о предмете из таблицы Predmet.

10. Остановить Microsoft SQL Server:

Пуск | Программы | Microsoft SQL Server | Service Manager ►

(ф. SQL Server Service Manager | кн. Stop ►

(ф. Are you sure ... ? | кн. Да) ► ... MS SQL Server – Stopped; кн.

[X] ►)

**Изучение возможностей программы Query Analyzer СУБД SQL Server
по созданию и ведению баз данных с использованием SQL**

1. Запустить Microsoft SQL Server:

Пуск | Программы | Microsoft SQL Server | Service Manager ►

(ф. SQL Server Service Manager | кн. Start/Continue ► ... MS SQL Server
– Running; кн. [X] ►)

2. Запустить Query Analyzer

Пуск | Программы | Microsoft SQL Server | Query Analyzer ►

ф. SQL Server Query Analyzer (ф. Connect to SQL Server | кн. OK) ►

ф. Query

3. Создать SQL-запрос на создание новой базы данных SSPP на Microsoft SQL Server:

ф. Query | ввести текст запроса:

```
CREATE DATABASE sspp_идентГруппы
```

– сохранить запрос:

ф. Query | кн. Save Query/Result ►

(ф. Save Query | Папка ← *имя папки*; Имя файла ← *Create_DB*; кн. Сохранить);

– создать новую базу данных, используя SQL-запрос:

ф. Query | кн. Execute Query (F5) ►

(окно Results | сообщение об успешном выполнении запроса);

DB ← имя созданной базы данных

– с помощью программы Enterprise Manager и убедиться наличии базы данных (см. предыдущую лабораторную работу).

4. Создать SQL-запрос Create_s на создание в базе данных структуры таблицы S (справочника поставщиков):

ф. Query | DB ← имя базы данных; ввести текст запроса:

```
CREATE TABLE s (s_no CHAR(4) NOT NULL, sname
VARCHAR(20) NOT NULL,
status NUMERIC, city VARCHAR(20))
```

– создать новую таблицу, используя SQL-запрос:

ф. Query | кн. Execute Query (F5) ►

(окно Results | сообщение об успешном выполнении запроса);

– сохранить запрос:

ф. Query | кн. Save Query/Result ►

(ф. Save Query | Папка ← *имя папки*; Имя файла ← *Create_s*; кн. Сохранить);

5. Аналогичным образом создать SQL-запрос Create_p на создание в базе данных структуры таблицы P (справочника товаров) и создать таблицу:

```
CREATE TABLE p (p_no CHAR(4) NOT NULL, pname VARCHAR(20)
NOT NULL,
color VARCHAR(10), weight NUMERIC, city VARCHAR(20))
```

6. Аналогичным образом создать SQL-запрос Create_sp на создание в базе данных структуры таблицы SP (сведений о поставках конкретных товаров конкретными поставщиками) и создать таблицу:

```
CREATE TABLE sp(s_no CHAR(4) NOT NULL, p_no CHAR(4) NOT
NULL, qty NUMERIC)
```

7. Создать SQL-запрос Alter_s на модификацию структуры таблицы S — ввести ограничение первичного ключа с именем PK_s_no по полю s_no — и модифицировать таблицу:

```
ALTER TABLE s ADD CONSTRAINT PK_s_no PRIMARY KEY (s_no)
```

8. Аналогичным образом создать SQL-запрос Alter_p на модификацию структуры таблицы P — ввести ограничение первичного ключа с именем PK_p_no по полю p_no — и модифицировать таблицу:

```
ALTER TABLE p ADD CONSTRAINT PK_p_no PRIMARY KEY (p_no)
```

9. Аналогичным образом создать SQL-запрос Alter_sp на модификацию структуры таблицы SP — ввести ограничение первичного ключа с именем PK_s_no_p_no по полям s_no и p_no, ограничение внешнего ключа FK_s_no по полю s_no для родительской таблицы s, ограничение внешнего ключа FK_p_no по полю p_no для родительской таблицы p — и модифицировать таблицу:

```
ALTER TABLE sp ADD  
    CONSTRAINT PK_s_no_p_no PRIMARY KEY (s_no, p_no),  
    CONSTRAINT FK_s_no FOREIGN KEY (s_no) REFERENCES s,  
    CONSTRAINT FK_p_no FOREIGN KEY (p_no) REFERENCES p
```

10. Создать SQL-запрос Insert_s на ввод 5 строк в таблицу S и заполнить ее:

```
INSERT INTO s VALUES ('s1', 'Smith', 20, 'London')  
INSERT INTO s VALUES ('s2', 'Jones', 10, 'Paris')  
INSERT INTO s VALUES ('s3', 'Blake', 30, 'Paris')  
INSERT INTO s VALUES ('s4', 'Clark', 20, 'London')  
INSERT INTO s VALUES ('s5', 'Adams', 30, 'Athens')
```

11. Аналогичным образом создать SQL-запрос Insert_p на ввод 6 строк в таблицу P и заполнить ее:

```
INSERT INTO p VALUES ('p1', 'Nut', 'Red', 12, 'London')  
INSERT INTO p VALUES ('p2', 'Bolt', 'Green', 17, 'Paris')  
INSERT INTO p VALUES ('p3', 'Screw', 'Blue', 17, 'Rome')  
INSERT INTO p VALUES ('p4', 'Screw', 'Red', 14, 'London')  
INSERT INTO p VALUES ('p5', 'Cam', 'Blue', 12, 'Paris')  
INSERT INTO p VALUES ('p6', 'Cog', 'Red', 19, 'London')
```

12. Аналогичным образом создать SQL-запрос Insert_p на ввод 12 строк в таблицу SP и заполнить ее:

```
INSERT INTO sp VALUES ('s1', 'p1', 300)
INSERT INTO sp VALUES ('s1', 'p2', 200)
INSERT INTO sp VALUES ('s1', 'p3', 400)
INSERT INTO sp VALUES ('s1', 'p4', 200)
INSERT INTO sp VALUES ('s1', 'p5', 100)
INSERT INTO sp VALUES ('s1', 'p6', 100)
INSERT INTO sp VALUES ('s2', 'p1', 300)
INSERT INTO sp VALUES ('s2', 'p2', 400)
INSERT INTO sp VALUES ('s3', 'p2', 200)
INSERT INTO sp VALUES ('s4', 'p2', 200)
INSERT INTO sp VALUES ('s4', 'p4', 300)
INSERT INTO sp VALUES ('s4', 'p5', 400)
```

13. Создать SQL-запрос Select_1a на извлечение данных из базы — «О поставщиках из Афин и Лондона, упорядоченных по статусу, а для одинакового статуса — по именам, используя: а) логическую функцию» — и извлечь данные:

```
SELECT * FROM s WHERE city='Athens' OR city='London' ORDER BY
status, sname
```

14. Аналогичным образом создать SQL-запросы Select_1b...Select12 на извлечение следующих данных из базы и извлечь данные:

1) О поставщиках из Афин и Лондона, упорядоченных по статусу, а для одинакового статуса — по именам, используя:

б) операцию IN:

```
SELECT * FROM s WHERE city IN ('Athens','London') ORDER BY status,
sname
```

2) О товарах: коды, имена, вес в килограммах:

```
SELECT p_no, pname, weight*0.001 AS kg FROM p
```

- 3) Коды поставщиков, поставляющих какие-либо товары (без дубликатов):

```
SELECT DISTINCT s_no FROM sp
```

- 4) О поставщиках, в именах или названиях городов которых есть сочетание букв «th»:

```
SELECT * FROM s WHERE sname LIKE "%th%" OR city LIKE "%th%"
```

- 5) О поставщиках со статусом от 20 до 30 (использовать BETWEEN):

```
SELECT * FROM s WHERE status BETWEEN 20 AND 30
```

- 6) О поставках с указанием кодов и имен поставщиков, кодов и имен товаров, количества товаров в поставке, используя для соединения таблиц:

- а) условие WHERE:

```
SELECT s.s_no,s.sname, p.p_no, p.pname, sp.qty FROM s, sp, p
WHERE s.s_no=sp.s_no AND sp.p_no=p.p_no
```

- б) оператор JOIN:

```
SELECT s.s_no, s.sname, p.p_no, p.pname, sp.qty
FROM (s INNER JOIN sp ON s.s_no=sp.s_no) INNER JOIN p ON
```

```
sp.p_no=p.p_no
```

- 7) О поставщиках хотя бы одного товара красного цвета, используя:

- а) вложенные запросы:

```
SELECT * FROM s WHERE s_no IN (
  SELECT s_no FROM sp WHERE p_no IN (
    SELECT p_no FROM p WHERE color = 'Red'))
```

- б) соединение таблиц:

```
SELECT DISTINCT s.* FROM s,sp,p
WHERE s.s_no=sp.s_no AND sp.p_no=p.p_no AND p.color='Red'
```

- 8) О поставщиках хотя бы одного товара, поставляемого поставщиком 's2', используя:

- а) вложенные запросы:

```
SELECT * FROM s WHERE s_no <> 's2' AND s_no IN (
    SELECT s_no FROM sp WHERE p_no IN (
        SELECT p_no FROM sp WHERE s_no = 's2'))
```

б) соединение таблиц:

```
SELECT DISTINCT s.* FROM s, sp spx, sp spy
WHERE s.s_no<>'s2' AND s.s_no=spx.s_no AND spx.p_no=spy.p_no
AND spy.s_no='s2'
```

9) О поставщиках, не поставляющих товары:

```
SELECT * FROM s WHERE s_no NOT IN (SELECT s_no FROM sp)
```

10) Коды и имена поставщиков, общее количество поставляемых поставщиком товаров sum_p, общее количество поставок поставщика kol_p, минимальное min_p, максимальное max_p и среднее sredn_p количество товаров в партиях поставщика, упорядоченные по общему количеству поставляемых товаров:

а) для поставщиков, поставляющих товары:

```
SELECT s.s_no, s.sname, SUM(qty) AS sum_p, COUNT(*) AS kol_p,
    MIN(qty) AS min_p, MAX(qty) AS max_p, AVG(qty) AS sredn_p
FROM s INNER JOIN sp ON s.s_no=sp.s_no GROUP BY s.s_no, s.sname
ORDER BY sname
```

б) для всех поставщиков (использовать внешнее соединение таблиц):

```
SELECT s.s_no, s.sname, SUM(qty) AS sum_p, COUNT(*) AS kol_p,
    MIN(qty) AS min_p, MAX(qty) AS max_p, AVG(qty) AS sredn_p
FROM s LEFT JOIN sp ON s.s_no=sp.s_no GROUP BY s.s_no, s.sname
ORDER BY SUM(qty) DESC
```

в) для всех поставщиков (использовать объединение таблиц):

```
SELECT s.s_no, s.sname, SUM(qty) AS sum_p, COUNT(*) AS kol_p,
    MIN(qty) AS min_p, MAX(qty) AS max_p, AVG(qty) AS sredn_p
FROM s INNER JOIN sp ON s.s_no=sp.s_no GROUP BY s.s_no, s.sname
UNION
```

```
SELECT s_no, sname, 0 AS sum_p, 0 AS kol_p, 0 AS min_p, 0 AS max_p,  
0 AS sredn_p  
FROM s WHERE status>10 AND s_no NOT IN (SELECT s_no FROM sp)  
ORDER BY sname
```

11) О поставщиках всех товаров:

```
SELECT * FROM s WHERE NOT EXISTS (  
SELECT * FROM p WHERE p_no NOT IN (  
SELECT p_no FROM sp WHERE s.s_no=sp.s_no AND sp.p_no=p.p_no))
```

12) О поставщиках по крайней мере тех товаров, которые поставяет поставщик 's2':

```
SELECT * FROM s WHERE s_no <> 's2' AND NOT EXISTS (  
SELECT * FROM sp WHERE s_no='s2' AND p_no NOT IN (  
SELECT p_no FROM sp WHERE s.s_no=sp.s_no))
```

Изучение возможностей доступа к СУБД SQL Server из других сред

А. Из среды Microsoft Access на основе использования проектов

1. Запустить Microsoft SQL Server.

2. Запустить Access. Создать новый проект:

Пуск | Программы | Microsoft Access | (ф. Microsoft Access | кн. Отмена); кн.

Создать ►

(ф. Создание | Общие | Проект (существующая база данных); кн. ОК) ►

(ф. Файл новой базы данных | нужная папка; Имя файла ← *sspp*; кн. Создать) ►

ф. Data Link Properties

– настроить связь через ADO для доступа к SQL Server:

ф. Data Link Properties | Connection | Select or enter server name ← (*local*) или имя компьютера;

уст. Use a specific user name and password; User name ← *sa*; уст. Blank password;

Select the database on the server ← *sspp*; (кн. Test connection ► Test connection succeeded);

кн. ОК ► ф. SSPP: база данных

3. Убедиться в работоспособности проекта, действуя аналогично доступу к обычным базам данных Access:

– открыть таблицы и просмотреть их содержимое;

– внести изменения в таблицы (добавление строк, изменение полей, удаление строк) и с помощью программы Enterprise Manager непосредственно убедиться в том, что изменения действительно внесены на сервер;

– создать новую таблицу на сервере в режиме конструктора.

Б. Из среды Microsoft Visual FoxPro на основе удаленных просмотров

1. Запустить Microsoft SQL Server.
2. Запустить Visual FoxPro. Создать новый проект. Создать новую базу данных SQLServer.
3. В базе данных создать новое ODBC-соединение с СУБД SQL Server:

ф. Project Manager | стр.Data | Databases | sqlserver | Connections; кн. New ►

(ф. Connection Designer

– создать новый источник данных ODBC для доступа к SQL Server:

в ф. Connection Designer | кн. New Data Source ►

(ф. ODBC Data Source Administrator | User DSN | кн. Add ►

(ф. Create New Data Source | Select a driver ... | выбрать SQL Server; кн. ГОТОВО ►)

(ф. Create New Data Source to SQL Server | Name ← *SQLServer_connection*;

Server ← (local); кн. Далее | Login ID ← *sa*; кн. Далее |

Change the default database to ← *sspp*; кн. Далее | кн. ГОТОВО



(ф. ODBC Microsoft SQL Server Setup | кн. Test Data Source



ДОЛЖНО БЫТЬ: TESTS COMPLETED
SUCCESSFULLY! кн. OK ►)

кн. OK ►)

кн. OK ►)

– создать соединение через созданный источник *SQLServer_connection*:

в ф. Connection Designer | Data Source ← *SQLServer_connection*; User ID ← *sa*;
Database ← *sspp*;

кн. Verify Connection ► должно быть (ф. Connection succeeded; OK)

кн. [X] ► (ф. Do You want to save... Yes ► (ф. Save | Connection Name ← *SQLServer_con*; OK ►)

4. Через новое ODBC-соединение создать новый удаленный просмотр для доступа к таблице s базы данных sspp СУБД SQL Server:

ф. Project Manager | стр.Data | Databases | sqlserver | Remote Views; кн. New ►
(ф. Select Connection or Data Source | уст. Connection; выбр. SQLServer_con; ОК)
(ф. View Designer |
(ф. Open | Tables in SQLServer_con | дв.клик 's'; кн. Close);
(стр. Fields | кн. ADD ALL>> ► перенос всех полей в Selected fields);
(стр. Update Criteria | уст. Send SQL Updates);
кн. [X]) ► (ф. Do you want to save changes? | кн.ОК) ►
(ф. Save | View Name ← s_SQLServer; кн.ОК) ►).

5. Убедиться в работоспособности удаленного просмотра:

ф. Project Manager | стр.Data | Databases | sqlserver | Remote Views |
's_sqlserver'; кн. Browse => содержимое таблицы S.

– убедиться в том, что модификация данных через удаленный просмотр (добавление, изменение и удаление строк таблицы) отражается на сервере: через удаленный просмотр внести изменения; через окно Command Window подать команду TABLEUPDATE(); через программу Enterprise Manager непосредственно обратиться к СУБД SQL Server, открыть таблицу s базы данных sspp и убедиться в наличии соответствующих изменений.

В. Из среды Microsoft Visual FoxPro на основе функций Pass-Through SQL

1. Запустить Microsoft SQL Server.
2. Запустить Visual FoxPro. Открыть предыдущий проект.
3. Создать новую программу:

ф. Project Manager | стр. Code | Programs | кн. New

– ввести текст:

```

h
SQLStringCONNECT("DSN=SQLServer_connection;Database=sspp;UID=sa;
PWD=")
IF h<0
    = AERROR(err)
    = MESSAGEBOX(err[2],16,"Ошибка соединения с SQL Server")
    RETURN
ENDIF
l = SQLExec(h, "SELECT * FROM s", "Cursor")
IF l < 0
    = AERROR(err)
    = SQLDISCONNECT(h)
    = MESSAGEBOX(err[2],16,"Ошибка выполнения команды на SQL
Server'e")
    RETURN
ENDIF
BROWSE
= SQLDISCONNECT(h)

```

– сохранить файл и выполнить программу.

**Изучение возможностей доступа к базам данных из Интернета
на примере Personal Web Server и СУБД Microsoft Access**

А. На основе технологии IDC/HTX

1. Создать базу данных Access SSPP как в лабораторной работе № 4 (достаточен упрощенный вариант без ограничений целостности: таблицы s (s_no, sname, status, city), p (p_no, pname, color, weight, city) и sp (s_no, p_no, qty), заполнить теми же данными, что и в работе № 4).
2. На Web-сервере опубликовать свою страницу с формой запроса, позволяющей пользователю задавать код поставщика, сведения о котором нужно извлечь из базы данных. Для этого в домашнем каталоге PWS (по умолчанию c:\interpub\wwwroot) разместить свою папку (например, Ivanov_IDC), в ней поместить файлы:

– global.asa:

```
<SCRIPT LANGUAGE=JSCRIPT RUNAT=Server>  
function Session_OnStart(){ }  
function Session_OnEnd(){ }  
function Application_OnStart(){ }  
function Application_OnEnd(){ }  
</SCRIPT>
```

– default.asp:

```
<html><head><title>Студенты группы АСУ-317a</title></head><body  
BGCOLOR="#ffffff">  
<table BORDER="0" CELLPADDING="5" CELLSPACING="0"> <tr><td>  
    <form ACTION="Query.idc" METHOD="post" TARGET="_top">  
    <h2>Технология IDC/HTX</h2><p>Доступ к БД SSPP:</p>  
    <p>запрос сведений о поставках </p>  
    <table BORDER="0" CELLPADDING="5" CELLSPACING="0">
```

```

        <tr><td>Код поставщика </td><td><nobr><input SIZE="5"
        NAME="s_no" value="s1"></td>
<td><input TYPE="submit" VALUE="Запросить"></nobr> </td></tr>
</table></form>
</td></tr></table></body></html>

```

Опробовать созданную страницу, для чего запустить Web-браузер (Internet Explorer) и ввести адрес //<имя компьютера>/<название страницы>/, например, //beta3/ivanov_idc/.

3. Создать источник данных ODBC с именем Access_SSPP на основе драйвера СУБД Microsoft Access для доступа к базе данных SSPP (аналогично тому, как это было сделано в лабораторной работе № 4). Доступ к Администратору ODBC:

кн. Пуск | Настройка | Панель управления | Источники данных ODBC

4. На Web-сервере опубликовать страницу, формирующую ответ на запрос пользователя (сведения о поставщике и его поставки). Для этого в папку поместить файл интернет-коннектора query.idc и файл HTML-шаблона query.htx:

– query.idc:

Datasource: Access_SSPP

SQLStatement: SELECT s.s_no,sname,sp.p_no,p.pname,sp.qty FROM s,sp,p
+WHERE s.s_no=sp.s_no AND sp.p_no=p.p_no AND sp.s_no='%s_no'

Template: Query.htx

– query.htx:

```
<html><head><title>Результаты запроса</title></head><body>
```

```
Поставщик <%idc.s_no%> <%sname%> <br>
```

```
Поставки:<br>-----<table>
```

```
<%begindetail%>
```

```
<tr><td><%p_no%><td><%pname%></td><td><%qty%></td></tr>
```

```
<%enddetail%>
</table></body></html>
```

Опробовать формирование ответа на запрос, вводя в форме (п. 2) различные значения кода поставщика и нажимая кнопку «Запросить».

Б. На основе технологии ASP/ADO и серверных сценариев JScript

5. Выполнить п. 2 для другой папки, например, Ivanov_ADO.

6. На Web-сервере опубликовать страницу, формирующую упрощенный ответ на запрос пользователя (сведения о поставщике без его поставок). Для этого в папку поместить файл с серверным сценарием query.asp:

```
<% @ LANGUAGE = "JScript" %>
<%
var cS_no=Request("s_no")(1);
var connect, rs, cQuery;
connect = Server.CreateObject("ADODB.Connection");
connect.Open("DSN=Access_SSPP");
cQuery = "SELECT * FROM s WHERE s_no='" + cS_no + "'"
rs = connect.Execute(cQuery)%>
<HTML><BODY><CENTER>
<H1>Запрос выполнен</H1>
<%if(rs.EOF){%>Несуществующий                поставщик
<%Response.Write(cS_no)}
else {%>Поставщик: <%=rs.Fields(0)%>. Имя: <%=rs.Fields(1)%>.
        Статус: <%=rs.Fields(2)%>. Город: <%=rs.Fields(3)%>. <%}
%>
</CENTER></BODY></HTML>
<%rs.Close();connect.Close();%>
```

Опробовать формирование ответа на запрос, вводя в форме различные значения кода поставщика и нажимая кнопку «Запросить».

7. Усложнить страницу query.asp так, чтобы она формировала полный ответ на запрос пользователя (сведения о поставщике и его поставках). Для этого в файл query.asp на место пустой строки поместить фрагмент:

```
cQuery = "SELECT p_no,pname,color,weight FROM p WHERE p_no IN "
+
      " (SELECT p_no FROM sp WHERE s_no='" + cS_no + "')";
rs = connect.Execute(cQuery);
if(rs.EOF){%><BR> * * * Нет поставок * * * <% }
else {%>
  <table border=1><tr>
    <td align="center"><b>Код</b></td>
    <td align="center"><b>Название</b></td>
    <td align="center"><b>Цвет</b></td>
    <td align="center"><b>Вес</b></td>
  </tr>
  <% while(!rs.EOF){%>
    <tr>
      <td align="center"><%=rs.Fields(0)%></td>
      <td align="center"><%=rs.Fields(1)%></td>
      <td align="center"><%=rs.Fields(2)%></td>
      <td align="center"><%=rs.Fields(3)%></td>
    </tr>
    <%rs.MoveNext();}%>
  </table>
  <% }
```

Опробовать формирование ответа на запрос, вводя в форме различные значения кода поставщика и нажимая кнопку «Запросить».