

что должно происходить при попытке ОБНОВЛЕНИЯ первичного ключа родительской таблицы, на которую ссылается некоторый внешний ключ?

что должно происходить при попытке УДАЛЕНИЯ первичного ключа родительской таблицы, на которую ссылается некоторый внешний ключ?

На эти вопросы можно дать три ответа (по выбору проектировщика):

0 КАСКАДИРОВАНИЕ. Операция обновления (или удаления) «каскадируется» с тем, чтобы обновить (удалить) также и внешний ключ в дочерней таблице. (На лабораторных работах для каскадирования обновления (и удаления) используется оператор CASCADE).

1 ОГРАНИЧЕНИЕ. Обновляются (или удаляются) первичные ключи лишь тех записей, на которые нет ссылок в дочерних таблицах. При этом используется оператор RESTRICT.

2 УСТАНОВЛЕНИЕ неопределенных значений (NULL-значений). Внешний ключ устанавливается в неопределенное значение, а затем обновляется первичный ключ родительской таблицы.

Целостность, определяемая пользователем. Для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к ней одной и определяются разработчиком. Чаще всего контролируется:

- уникальность тех или иных атрибутов,
- диапазон значений (обычно используется для числовых полей, различают открытые и закрытые диапазоны; первые фиксируют значение только одной из границ, а вторые фиксируют обе границы; например, экзаменационная оценка от 2 до 5;
- принадлежность набору значений или *заданию домена* (пол "М" или "Ж").

Ограничения целостности – это утверждения о допустимых значениях отдельных информационных единиц и связях между ними.

Ограничение целостности устанавливает правила на уровне баз данных, определяя набор проверок для таблиц системы. Эти проверки автоматически выполняются всякий раз, когда вызывается оператор вставки, модификации или удаления данных в таблице. Если какие-либо ограничения нарушены, операторы отменяются.

Для полей в таблицах, входящих в структуру базы данных, в основном используются следующие виды ограничений.

Тип и формат поля.

Признак непустого поля. Характеризует недопустимость пустого значения поля в БД. Например, поля со сведениями о сотрудниках («фамилия», «имя», «отчество», «оклад») обязательно должны иметь какое-то значение, а в поле «ученая степень» может отсутствовать значение.

В том случае, если речь идет об ограничениях целостности, относящихся к кортежу, то имеется в виду ограничение на значение всей

строки, рассматриваемой как единое целое, или ограничения на соотношения значений отдельных полей в пределах одной строки. Естественным ограничением является требование уникальности каждой строки таблицы.

По определению в реляционном отношении не может быть одинаковых кортежей. Однако не все реляционные СУБД обеспечивают автоматическое соблюдение этого ограничения. Пример ограничения на соотношения полей внутри одного кортежа: значение поля «дата возврата» не должно превышать «дату заказа».

Если встроенные возможности СУБД не позволяют контролировать подобные ограничения целостности, то надо написать универсальную программу (создать процедуру), позволяющую это делать.

Видом ограничения является также *запрет на обновление*. Он может относиться как к отдельному полю, так и к записи или целой таблице. В некоторых СУБД существует запрет на корректировку ключевого поля.

Запрет на обновление может быть связан с технологией обработки данных или со спецификой предметной области. Например, если описывается объект «Личность», то его атрибуты дата и место рождения являются постоянными и не могут меняться. Для соответствующих полей в БД рекомендуется задавать запрет на обновление. При этом запрет на обновление может относиться как к отдельному полю, так и ко всей записи и или файлу.

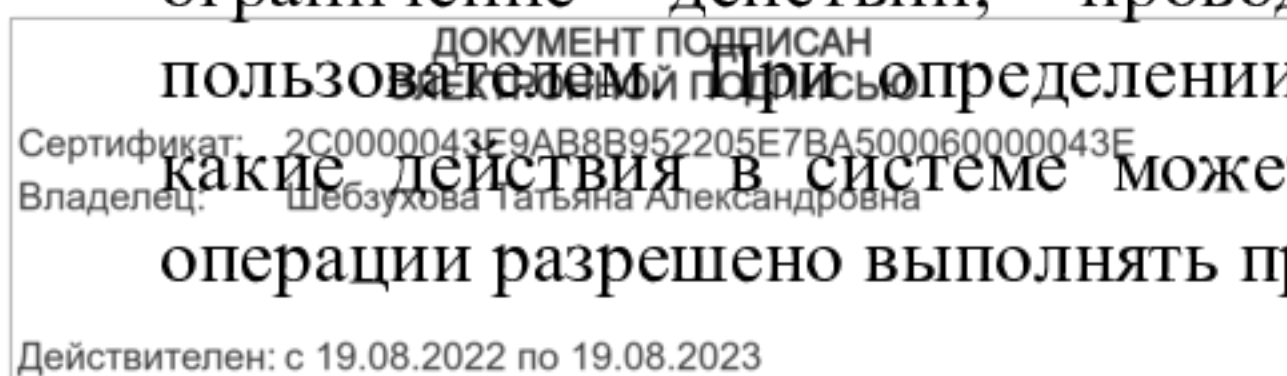
3.3.7 Запросы к базе данных

В шестом подразделе практической части курсового проекта по дисциплине «Управление данными» следует привести не менее 10 запросов всех типов, реализуемых средствами СУБД и средствами языка SQL. Составить запросы с коррелированными и некоррелированными подзапросами. Предварительно они должны быть сформулированы на естественном языке и выражены в терминах реляционной алгебры или реляционного исчисления. Примеры возможных запросов к базе данных и способы их реализации приведены в приложении 3.

3.3.8 Разработка механизмов защиты данных от несанкционированного доступа

При разработке механизмов защиты данных в разрабатываемой базе необходимо проанализировать состав обслуживающего персонала, который будет работать с ней. Рассмотреть привилегии, предоставляемые пользователям для работы с базой данных, таблицами, представлениями.

Целью управления доступом к объектам базы данных является ограничение действий, проводимых зарегистрированным в системе пользователем. При определении прав доступа администратор БД решает, какие действия в системе может выполнять пользователь, а также какие операции разрешено выполнять приложениям, запущенным от имени



пользователя. Управление доступом предназначено для предотвращения действий пользователя, которые способны нанести вред базе данных.

Под управлением доступом понимается возможность субъекта (сущность, определяющая пользователя при работе в системе) проводить действия над объектом (как базы данных целиком, так и ее компонентов). Различаются три метода управления доступом: дискреционный, обязательный, ролевой.

Дискреционный контроль предполагает, что владелец объекта сам может указывать, кто имеет доступ к объекту и вид этого доступа.

Для управления доступа к объектам дискреционный контроль доступа использует идентификационную информацию объекта и список доступа, который содержит субъекты и ассоциированные с ними типы доступа. Во время запроса доступа к объекту, система производит поиск субъекта в списке прав доступа объекта, и, в случае, если субъект там присутствует и разрешённый тип доступа включает требуемый тип, разрешает доступ. В противном случае запрещает доступ.

Благодаря своей гибкости, дискреционный контроль доступа широко используется. Одним из значительных недостатков данного вида контроля является отсутствие полной гарантии того, что информация не станет доступна другим субъектам, которые не имеют к ней доступа. Причина этого кроется в том, что субъект, имеющий право чтения информации может без уведомления владельца объекта передать её другим субъектам, не имеющим такого права. Дискреционная модель контроля доступа не накладывает ограничений на дальнейшее распространение информации после того, как субъект её получил.

Помимо этого, к недостаткам можно отнести ещё одну особенность дискреционной модели контроля доступа: объекты в системе принадлежат субъектам, которые настраивают доступ к ним для других. Но на практике, в большинстве случаев, данные в системе принадлежат всей системе, а не отдельным субъектам. Информационная система является наиболее распространённым примером.

Различают закрытую и открытую систему дискреционного контроля. Под закрытой системой понимается такая система, в которой изначально объект никому не доступен и список разрешений описывается в списке прав доступа, а открытая – та, в которой к объектам все имеют полный доступ, а в списке доступа описывается список ограничений.

Обязательный контроль доступа осуществляет управление доступом, опираясь на элементы самой системы. Каждому из элементов системы назначается уровень безопасности, который описывает важность этого объекта и ущерб, причинённый при разглашении информации в этом объекте. Уровень доверия субъекту является уровнем его безопасности. Все уровни безопасности являются членами определённой иерархии, то есть каждый уровень безопасности включает себя и уровни, находящиеся ниже.

Доступ субъекта к структурному элементу базы данных (таблице, запросу и т. д.), а также самой базе данных предоставляется в случае выполнения определённого условия отношения между уровнями безопасности субъекта и объекта. Доступ на выполнение операции даётся при выполнении определённых условий. Доступ на чтение даётся в случае, если уровень безопасности субъекта включает в себя уровень безопасности объекта. Доступ на запись даётся в случае, если уровень безопасности субъекта включается в уровень безопасности объекта. При выполнении этих условий гарантируется, что данные высокоуровневых объектов не попадут в низкоуровневые объекты.

Однако в такой модели существуют два момента, ставящие под вопрос её непротиворечивость.

0 Пользователь нижнего уровня может записывать информацию в объекты верхних уровней. Он может переписать существующий объект своим собственным и информация будет потеряна. Такой недостаток можно устранить, запретив запись на более верхние уровни. При такой схеме доступ на чтение будет даваться, если уровень безопасности субъекта включается в себя уровень безопасности объекта, а доступ на чтение будет даваться в случае, если уровень безопасности субъекта будет равняться уровню безопасности объекта.

1 Пользователи с более высоким уровнем не могут изменять объекты с более низки уровнем. Этот недостаток можно устранить, позволив пользователю при доступе к объектам выступать от имени субъектов с различными уровнями.

Часто, кроме обеспечения безопасности, также требуется обеспечение её достоверности. Чем выше уровень доверия объекта, тем выше его достоверность. Чем выше уровень безопасности субъекта, тем достовернее информация, которую он вносит в систему. Для такой модели описанные выше правила должны быть изменены: доступ на запись даётся в случае, если уровень безопасности субъекта включает в себя уровень безопасности объекта, а доступ чтение даётся в случае, если уровень безопасности субъекта включается в уровень безопасности объекта. То есть критерии просто поменялись местами.

Вместе с использованием уровней безопасности можно использовать категории. В таком случае, кроме уровня безопасности, каждому объекту и субъекту можно назначить список категорий, к которым он относится. Категории объекта используются для описания тех областей, в которых он используется. Категории субъекта описывают области, в которых он работает. Такая система позволит детальнее управлять доступом в системе.

классических моделях разграничения доступа права на выполнение определённых операций над объектом должны быть прописаны для каждого пользователя или группы пользователей. Благодаря разделению понятий «роль» и «пользователь», можно разбить задачу на две части: определение роли пользователя и определение прав доступа к объекту для роли. Такой

подход позволяет упростить процесс администрирования, так как при изменении области ответственности пользователя, требуется лишь убрать у него старые роли и назначить новые, соответствующие его текущим обязанностям. Эта же процедура потребует массу усилий при переназначении новых прав, если права доступа определялись напрямую между пользователями и объектами.

Посредством построения иерархии ролей, систему ролей можно настроить так, чтобы она точнее отображала реальные бизнес процессы. Каждая роль, вместе со своими привилегиями, может ещё наследовать привилегии других ролей. Это тоже упростит администрирование системы

Ролевая модель позволяет пользователю регистрироваться в системе с наименьшей ролью для выполнения требуемых задач. Пользователям, у которых множество ролей, не всегда требуются все привилегии для выполнения определённой задачи.

Принцип наименьшей привилегии обеспечивает достоверность данных в системе. Необходимо давать пользователю только те разрешённые привилегии, которые ему нужны для выполнения определённой задачи. Для этого следует выяснить цель задачи, привилегии, необходимые для её выполнения и ограничить этим набором привилегии пользователя. Запрещение привилегий, не требуемых для выполнения текущей задачи, не позволит обойти политику безопасности системы.

Разделение обязанностей также является одним из важных принципов в системе управления доступом. Не редки ситуации, когда ряд определённых действий не может быть выполнен одним и тем же человеком для предотвращения мошенничеств, например, операции создания и подтверждения платежа. Система ролевого управления доступом позволит решить эту задачу без особых затруднений.

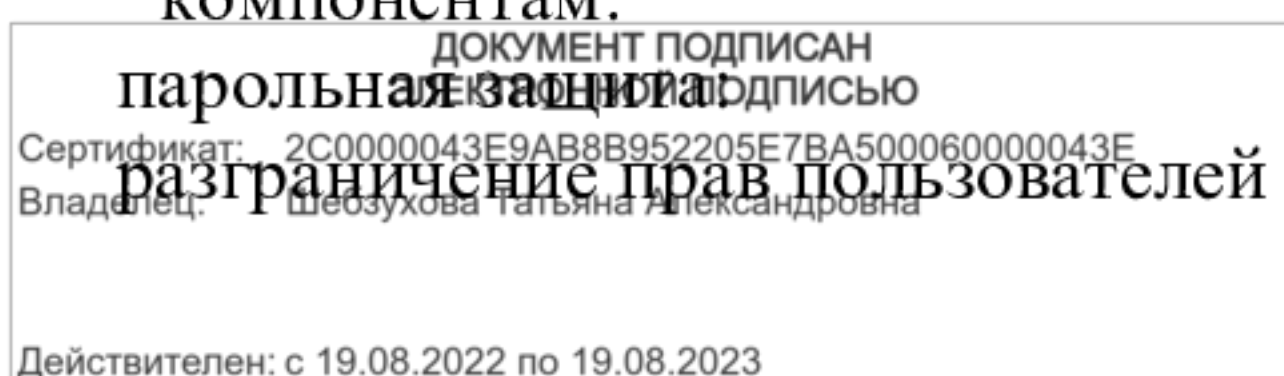
Модель состоит из сущностей пользователя, роли и привилегии, где пользователь – человек либо программа запущенная им, роль – вид деятельности человека в организации, а привилегия – разрешение доступа к объектам системы. У пользователя может быть несколько ролей, а одна роль может принадлежать нескольким пользователям. Также и несколько привилегий могут принадлежать одной роли, а несколько ролей – иметь одну привилегию.

Сессия, которую активизирует пользователь для выполнений задачи, позволяет отнести пользователя к множеству ролей. В этот момент система определяет роли и привилегии, требующиеся пользователю для выполнения задачи, и запрещает все остальные.

В проектируемой базе данных «Прокат» можно рассматривать следующие механизмы доступа как к базе данных целиком, так к ее компонентам:

парольная защита

разграничение прав пользователей (ролевая защита).



Пользователями базы данных «Прокат» могут быть менеджер первого звена, менеджер по работе с клиентами и администратор базы данных.

Из всех перечисленных пользователей наиболее широким спектром полномочий обладает администратор, который может выполнять все операции с базой данных и ее компонентами.

3.3.9 Требования к техническому обеспечению

В этом разделе следует сформулировать требования к системе БД, например, определить быстродействие системы, требования к аппаратному обеспечению, объему оперативной памяти и жесткого диска.

При описании технического обеспечения базы данных необходимо соблюдать следующие требования:

- к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию;
- к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения системы.

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующие в фирме средства компьютерной и вычислительной техники.

Для эффективного функционирования базы данных «Прокат» необходимы персональные компьютеры, являющиеся рабочими местами менеджера первого звена, менеджеров по работе с клиентами и администратора, которые могут обладать следующими техническими характеристиками:

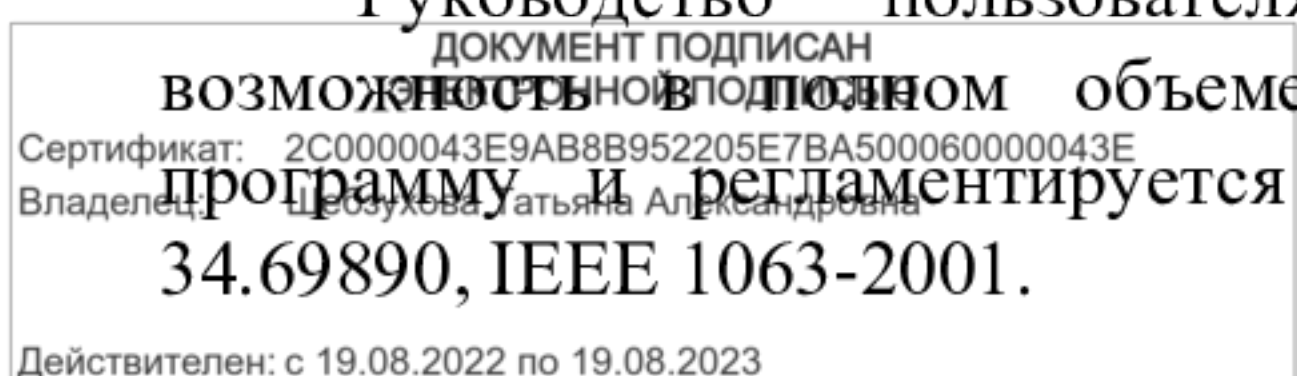
- процессор – Intel Core i3 380M;
- объем оперативной памяти – 6 Гб;
- жесткий диск объемом не менее 500 Гб;
- дисковая подсистема – 40 Гб;
- устройство чтения компакт-дисков (DVD-ROM); - сетевой адаптер – 100 Мбит.

3.3.10 Инструкция по использованию базы данных

При разработке инструкции для пользователя базы данных необходимо описать, как производится вызов программы, и как характеризуются ее основные функции. В этом разделе приводятся справочные сведения о разработанной базе данных, описание экранных форм и отчетов.

Руководство пользователя — один из основных программных документов. Основная задача документа состоит в том, чтобы обеспечить пользователям возможность самостоятельно решать все основные задачи, на которые нацелена программа.

Руководство пользователя должно обеспечить пользователям возможность в полном объеме самостоятельно освоить и применять программу и регламентируется стандартами ГОСТ 34.201-89, РД 50-34.69890, IEEE 1063-2001.



Руководство пользователя содержит полное описание программы с точки зрения ее целевого применения. В руководстве пользователя обязательно должны быть описаны:

назначение базы данных;

основные задачи и возможности;

способ отражения предметной области в базе

данных; пользовательский интерфейс базы данных;

порядок решения основных пользовательских задач;

все функции программы и порядок их применения;

пользовательская настройка программы; проблемы

при использовании и способы их решения.

При документировании небольших программ в руководство пользователя часто включают инструкции по установке, настройке, администрированию, обновлению и прочему обслуживанию программы.

зависимости от особенностей программы и целевой аудитории руководство пользователя по способу изложения материала может приближаться к учебнику или, наоборот, к справочнику. Порядок изложения материала в руководстве пользователя определяется *пользовательской перспективой* программы.

Если программа представляет собой инструмент, позволяющий решать практические задачи из некоторого конечного набора, то в руководстве приводят типовые процедуры решения каждой из них. Каждое из этих действий можно разложить на последовательные элементарные шаги, во всяком случае, для типичных ситуаций. Руководство пользователя, построенное по принципу пользовательских задач, напоминает учебник, хотя, как правило, лишено присущего учебникам методического аппарата: проверочных заданий, вопросов, упражнений.

Если программа представляет собой среду, в пределах которой пользователь может решать задачи, поставленные им самостоятельно, руководство пользователя должно быть ближе к справочнику.

Если программа является инструментом, с помощью которого пользователь контролирует состояние того или иного объекта, например, промышленной установки, то руководство пользователя строится по принципу таблицы: сообщение программы - реакция или возможные реакции пользователя.

Если пользователь применяет программу для решения задач в нетривиальных предметных областях, в руководство пользователя настоятельно рекомендуется включить концептуальный раздел. В нем должен быть описан реализованный в программе способ представления объектов реального мира, чтобы пользователь хорошо понимал, с какими из них и на каком уровне абстракции он может работать.

В каждом конкретном случае структура руководства будет в основном определяться особенностями описываемой программы. Обычно структура руководства пользователя содержит следующие разделы:

Общие сведения.

Установка и первоначальная настройка.

Основные понятия и определения.

Интерфейс пользователя.

Работа с программой.

Пользовательская настройка.

Сообщения об ошибках.

Единый раздел «Работа с программой» часто заменяют несколькими последовательными разделами, описывающими крупные группы пользовательских задач или функций.

3.3.11 Заключение

Заключение к курсовому проекту должно содержать выводы о проделанной работе. Примерный объем заключения 1 – 2 страницы.

3.3.12 Список используемой литературы

В списке используемой литературы приводятся данные об литературных источниках информации (монографии, учебники, статьи, справочники и т. д.), а также ссылки на электронные издания по тематике курсовой работы и Internet-адреса соответствующих сайтов. Содержание списка нумеруется по мере упоминания в пояснительной записке. Пример оформления литературных источников приведен в разделе «Оформление пояснительной записки».

3.3.13 Приложения Пояснительная записка к курсовому проекту по дисциплине

«Управление данными» может содержать перечисленные приложения:

словарь терминов, используемых в данной предметной области;

формы входных документов;

формы выходных документов;

экранные формы;

распечатки реализованных запросов;

распечатки реализованных отчетов; - тексты программ.

ОБЩИЕ ТРЕБОВАНИЯ К НАПИСАНИЮ И ОФОРМЛЕНИЮ

РАБОТЫ Выполнение курсовой работы и подготовка его к защите включает этапы:

- 0 выбор и оформление темы курсовой работы;
- 1 разработка и утверждение задания на курсовую работу;
- 2 реализация этапов проектирования информационной подсистемы;
- 3 оформление пояснительной записки к курсовой работе.

Тематика курсовых работ должна быть актуальной, соответствовать современному состоянию и перспективам развития информационных систем средств сбора, передачи и отображения информации в различных предметных областях. При выборе темы необходимо учитывать следующие

положения:

0 Актуальность проблемы и ее значимость для практической деятельности базовой организации.

1 Интересы, склонности, имеющийся задел научно-исследовательской работы студента во время предшествующего обучения, а также перспектив его будущей профессиональной деятельности.

2 Возможность использования полученных результатов и дальнейшего развития темы при выполнении дипломного проектирования.

3 Наличие специальной научной литературы для теоретического обоснования проблемы.

Тема курсового проекта должна быть краткой, отражать его основное содержание. В названии темы должны быть указаны область деятельности и объект, на которые ориентирован проект. Закрепление темы курсового проекта осуществляется по предоставлению кафедры и утверждается приказами по директорату. Темы курсовых работ (проектов) утверждаются на заседании кафедры, ведущей дисциплины, по которым учебными планами предусмотрены курсовые работы (проекты), в течение 2-х недель после начала семестра. Выписка из протокола заседания кафедры передается дирекцию института, где формируется распоряжение об утверждении тем курсовых работ (проектов). В течение 10 дней после выхода распоряжения об утверждении тем курсовых работ (проектов) специалист по учебно-методической работе дирекции вносит темы работ в автоматизированную базу университета.

После этого руководителем разрабатывается задание на курсовой проект, включающее аналитическую и практическую часть разработки, исходные данные, описывающие предметную область для которой нужно разработать локальную базу данных.

Руководитель проекта регулярно, не реже одного раза в месяц, проводит консультации, на которых рекомендует студенту необходимую литературу, справочные и методические материалы, стандарты и типовые решения, а также проверяет ход ее выполнения, сохраняя за студентом полную самостоятельность.

Задание на курсовой проект оформляется в соответствии с требованиями, утвержденными в ВУЗе (приложение 1). Титульный лист оформляется по образцу, приведенному в приложении 2.

4.1 Оформление пояснительной записки

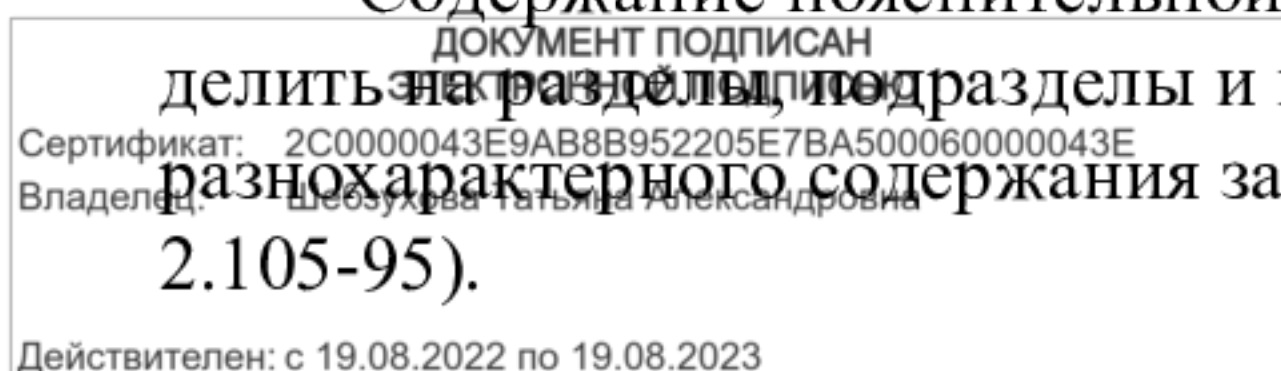
Пояснительная записка к курсовому проекту относится к текстовым документам и должна соответствовать требованиям:

ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам;

ГОСТ 2.106-96 ЕСКД «Текстовые документы»

Содержание пояснительной записки к курсовому проекту следует

делить на разделы, подразделы и пункты. В случае значительного объема и разнохарактерного содержания записку следует делить на части (ГОСТ 2.105-95).



детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, запись производится с абзацного отступа, как показано на примере.

Пример

а) _____

б) _____

в) _____

Каждый пункт, подпункт и перечисление записывают с абзацного отступа.

Разделы, подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов, подразделов. Заголовок следует печатать с прописной буквы без точки в конце, не подчеркивая. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой. Расстояние между заголовком и текстом при электронном наборе должно составлять два полуторных интервала. Каждый раздел текстового документа рекомендуется начинать с нового листа (страницы).

Слово «Содержание» записывают в виде заголовка (симметрично тексту) с прописной буквы. Наименования, включенные в содержание, записывают строчными буквами, начиная с прописной буквы.

В конце текстового документа перед листом регистрации изменений допускается приводить список литературы, которая была использована при его составлении. Выполнение списка и ссылки на него в тексте – по ГОСТ 7.32. Список литературы включают в содержание документа.

Нумерация страниц документа и приложений, входящих в состав этого документа, должна быть сквозная.

Текст пояснительной записки должен быть кратким, четким и не допускать различных толкований.

При изложении обязательных требований в тексте пояснительной записки необходимо применяться слова «должен», «разрешается только», «следует», «необходимо», «требуется чтобы», «не допускается», «запрещается», «не следует». При изложении в ПЗ других положений следует применять слова – «могут быть», «как правило», «при необходимости», «может быть», «в случае» и т.д.

Формулы в тексте пояснительной записки нумеруют сквозной нумерацией арабскими цифрами, например, (1), (2) и т. д. Допускается нумеровать формулы в пределах раздела. Формулы в приложении, например приложении А, нумеруют следующим образом (А.1), (А.2) и т. д.

Допускается нумерация формул в пределах раздела, например (3.1) – раздел 3 формула 1. Ссылки на формулы в тексте пояснительной записки

рекомендуется выполнять следующим образом:...как показано в формуле (5) ...;...как следует из соотношений (2) – (6) ...

Примечания помещают непосредственно после текста, графического материала или таблиц, к которым относятся эти примечания, и печатают с прописной буквы с абзаца. Если примечание одно, то после слова «Примечание» ставится тире и примечание тоже печатается с прописной буквы. Одно примечание не нумеруется. Несколько примечаний нумеруют по порядку арабскими цифрами. Примечания к таблице помещают в конце таблицы под линией, обозначающей окончание таблицы.

Иллюстрации могут быть размещены как по тексту пояснительной записки, так и в конце нее. Иллюстрации следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается Рисунок 1. Иллюстрация в приложении обозначается Рисунок А.3 (приложение А, рисунок 3). Допускается нумеровать иллюстрации в пределах раздела, например, Рисунок 1.1 (первый раздел, рисунок один), Рисунок 3.5 (третий раздел, рисунок пять). Ссылки на рисунки в тексте пояснительной записки выполняются следующим образом: ...в соответствии с рисунком 2....., ...в соответствии с рисунком 1.2.... Подрисуночный текст оформляется следующим образом: Рисунок 1 – Детали прибора В конце подрисуночного текста точка не ставится.

Приложения в пояснительной записке являются информационными (носят справочный характер). Каждое приложение начинается с новой страницы с указанием наверху посередине страницы слова «Приложение» и его обозначения. Под ним в скобках для информационного приложения пишут слово «рекомендуемое» или «справочное». Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь. После слова «Приложение» следует буква, обозначающая его последовательность. Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O. В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами. Если в пояснительной записке одно приложение, оно обозначается «ПРИЛОЖЕНИЕ А». Приложения выполняются на листах формата А4 (допускаются форматы А3, А4×3, А4×4, А2 и А1).

Цифровой материал оформляют в виде таблиц. Пример оформления таблицы приведен ниже.

Графу «Номер по порядку» в таблицу включать не допускается. Нумерация граф таблицы арабскими цифрами допускается в случаях: 1) когда в тексте есть ссылки на них; 2) при делении таблицы на части; 3) при переносе части на следующую страницу. Если в тексте пояснительной записки одна таблица, она обозначается «Таблица 1» или «Таблица А.1» (в

приложении А). Допускается нумерация таблиц в пределах раздела, например, «Таблица 1.2».

Ссылки на таблицы в тексте пояснительной записки: ... как показано в таблице 1..., ... как следует из таблицы 2.5... Таблица 5 – Сводная ведомость

(конец страницы)

.....

Продолжение таблицы 5

--	--	--

5. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЯ

Содержание основных этапов подготовки курсового проекта по дисциплине «Управление данными» представлено в таблице 5.1. Таблица 5.1 – Основные этапы подготовки курсового проекта

Наименование этапа работы	Срок выполнения
Получение задания на курсовое проектирование	2 неделя семестра
Выполнение теоретического задания	3-9 недели семестра
Предварительное обследование предметной области и оформление его результатов	5 неделя семестра
Составление технического задания на разработку базы данных	6 неделя семестра
Инфологическое проектирование	7 неделя семестра
Преобразование ER-модели в реляционную	8 неделя семестра
Даталогическое проектирование, загрузка базы данных, тестирование и отладка	9 неделя семестра
Разработка запросов	10-12 недели семестра
Создание справочной системы	13 неделя семестра
Создание входных и выходных форм	14 неделя семестра
Оформление пояснительной записки	15 неделя семестра
Защита проекта	17 неделя семестра

ДОКУМЕНТ ПОДПИСАН

Сертификат: 2C00000043E9AB8B952205E7BA50000600000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

6. КРИТЕРИИ ОЦЕНИВАНИЯ ПРОЕКТА

Оценка «отлично» при защите курсового проекта выставляется в том случае, если он свободно ориентируется в исследуемой предметной области может дать четкие развернутые ответы на все поставленные преподавателем вопросы.

Оценка «хорошо» при защите курсового проекта может быть выставлена, если студент хорошо ориентируется в предметной области, рассмотренной в курсовом проекте, и может дать ответы на поставленные вопросы.

Оценка «удовлетворительно» может быть выставлена, если студент плохо ориентируется в предметной области, для которой выполнен курсовой проект, и им были даны ответы не на все вопросы, сформулированные в ходе защиты проекта, преподавателем.

Оценка «неудовлетворительно» при защите курсового проекта может быть выставлена, если:

0 студент вообще не ориентируется в той предметной области, для которой выполнен курсовой проект;

1 не может дать ответы на вопросы, заданные преподавателем;

2 две курсовые работы выполнены на одну и ту же тему.

Если студент на защите получает оценку «неудовлетворительно», то тема курсового проекта изменяется. Изменение темы курсового проекта происходит и в том случае, когда несколько пояснительных записок абсолютно одинаковы.

7. ПОРЯДОК ЗАЩИТЫ ПРОЕКТА

Пояснительная записка сдается на проверку руководителю работы в срок не менее чем за 10 дней до защиты. После проверки руководитель либо допускает студента к защите, либо возвращает проект на доработку.

Программная реализация обязательно прилагается на диске и демонстрируется руководителю.

Порядок защиты курсового проекта и состав комиссии утверждается на заседании кафедры. Студент делает доклад (около 5 минут), в котором кратко излагает результаты проектирования, демонстрирует графическую часть проекта и разработанную программную реализацию.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Основная литература

Действителен: с 19.08.2022 по 19.08.2023

ПРИЛОЖЕНИЕ 1

УТВЕРЖДАЮ
Заведующий кафедрой

_____ (название кафедры) _____
(ФИО)

Институт _____
Кафедра _____
Направление (специальность) _____

Профиль (специализация) _____

ЗАДАНИЕ на курсовую работу (проект)

студента _____
(фамилия, имя, отчество)

по дисциплине _____

1 Тема работы

2 Цель

3 Задачи

—

Перечень подлежащих разработке вопросов:

а) по теоретической части _____

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 2500900043E9A58B332269E7BA30000000043E Владелец: Шебзухова Татьяна Александровна Действителен: с 19.08.2022 по 19.08.2023	б) по аналитической части _____ _____ _____
--	---

Исходные данные:

а) по литературным источникам _____

б) по вариантам, разработанным преподавателем _____

в)

иное _____

6. Список рекомендуемой литературы

Контрольные сроки представления отдельных разделов курсового проекта:

25 % - _____ “ ” _____

20_ г.

50 % - _____ “ ” _____ 20_ г.

75 % - _____ “ ” _____ 20_ г. 100

% - _____ “ ” _____ 20_ г.

8. Срок защиты студентом курсового проекта “ ” _____ 20_ г.

Дата выдачи задания “ ” _____ 20_ г.

Руководитель курсового проекта _____

(ученая степень, звание) (личная подпись) (инициалы, фамилия)

Задание принял(а) к исполнению студент(ка) _____ формы обучения

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебухова Татьяна Александровна

курса группы

(личная подпись)

(инициалы, фамилия)

Действителен: с 19.08.2022 по 19.08.2023

ПРИЛОЖЕНИЕ 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ И
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ
КАФЕДРА Систем управления и информационных технологий

КУРСОВОЙ ПРОЕКТ по дисциплине
«Управление данными» на тему:

« _____ »

Выполнил:

студент _____ курса группы _____ направления _____
_____ очной формы обучения

Фамилия, имя, отчество _____

(подпись)

Руководитель работы:

_____ (ФИО, должность,
кафедра)

Работа допущена к защите _____
(подпись руководителя) (дата)

Работа выполнена и защищена с _____ оценкой _____ Дата
защиты _____

Члены комиссии: _____
(должность) (подпись) (И.О. Фамилия)

Пятигорск, 20 ____ г.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат:	2C0000043E9AB8B952205E7BA500060000043E
Владелец:	Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023	

ПРИЛОЖЕНИЕ 3

1 Безусловная выборка значений

СУБД Visual FoxPro вводить SQL-запросы можно непосредственно в командном окне (Command window) или в окне дизайнера запросов (Query Designer). Для формирования запросов на языке **SQL** используется конструкция **SELECT**. Результатом выполнения запроса является таблица, которая хранится во временном буфере базы данных. Выбранные данные можно использовать для просмотра, формирования графиков или печати отчетов. Длина строки на языке **SQL** до 255 символов, регистр не имеет значения. Синтаксис команды **SELECT**:

SELECT [ALL|DISTINCT] список выбираемых полей

FROM список таблиц

[WHERE условие выборки или соединения]

[GROUP BY список полей по условию группировки

[HAVING условие выборки группы]]

[ORDER BY список полей, по которым упорядочить вывод]

При формировании запросов можно использовать уточненные имена полей (например, CUSTOMER.CUSTOMERNO, т.е. Имя_таблицы.Имя_поля).

Ключевое слово **ALL** подразумевается по умолчанию.

Для выборки всех полей таблицы в том же порядке, что и в таблице, используется следующая форма запроса:

SELECT*FROM таблица.

Запрос вида

SELECT*FROM R1, R2

соответствует декартову произведению таблиц **R1** и **R2**, т. е **R=R1x R2**.

Запрос

SELECT R1.A, R2.B FROM R1, R2

соответствует проекции декартова произведения таблиц **R1** и **R2** на столбцы из таблицы **R1** и **B** – из таблицы **R2**, то есть **R = R1.A, R2.B(R1x R2)**. 2

Простая выборка

При использовании механизма простой выборки предполагается, что в результате ее выполнения на экран будет выведен некоторый диапазон значений. Например, результатом выполнения запроса

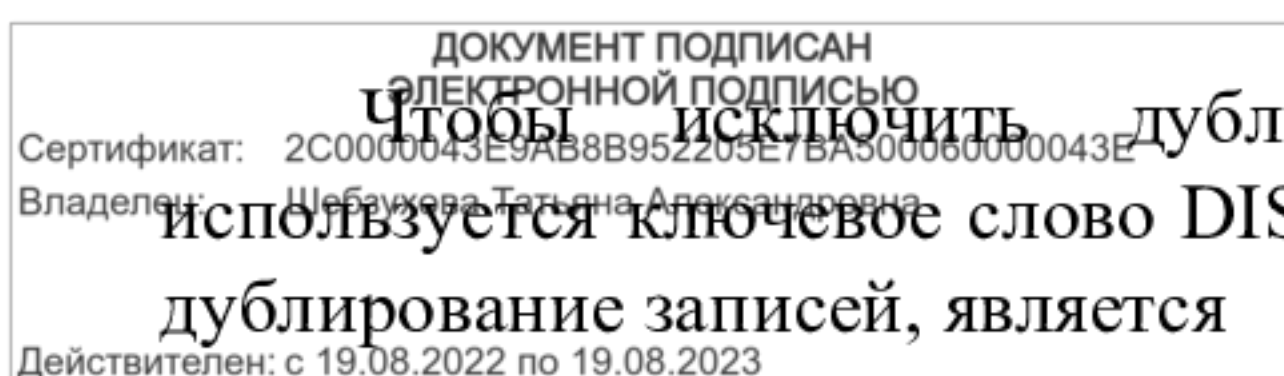
SELECT kod FROM R1

является столбец **kod** из таблицы **R1**.

При выполнении данного запроса в результат выборки будут включены все дубликаты строк.

3 Выборка уникальных значений

Чтобы исключить дубликаты строк из результата выборки используется ключевое слово **DISTINCT**. Примером запроса, исключающего дублирование записей, является



SELECT DISTINCT kod FROM R1

4 Выборка вычисляемых значений

При выполнении запросов в СУБД **Visual FoxPro** может осуществляться не только выбор из таблицы ранее введенных значений, но и получение данных, отсутствующих в исходной таблице. Для выборки вычисляемых значений в **Visual FoxPro** имеются встроенные функции и арифметические операторы. При организации выборки информации вычисления могут производиться по одному или нескольким полям исходной таблицы.

Чтобы включить в запрос функцию поля или выражение, необходимо:

в окне конструктора запросов активизировать вкладку **Fields**;

с помощью манипулятора «мышь» или клавиши **Tab** перейти в поле

«**Functions and expressions**»

в этом поле ввести выражение для вычисления или нажать кнопку вызова построителя выражения, расположенную с правой стороны поля,

в окне диалога **Expression Builder** создать выражение для вычисляемого поля.

нажать кнопку **Add** для переноса данного выражения в список выходных полей запроса.

Пример 4.1. Пусть в некоторой базе данных содержится таблица **Товары**, в которой цена (**UNITPRICE**) указана без учёта налога на добавленную стоимость (НДС). Поля таблицы **Товары** представлены в таблице 4.1. Тип переменной **UNITPRICE** – **Currency**.

Таблица 4.1 – Поля таблицы **Товары**

Kod1	UNITPRICE
1000.0000	
2000.0000	
310000.0000	

Необходимо организовать выборку информации из таблицы **Товары**, указав цены с учетом налога на добавленную стоимость (18%). SQL-запрос, соответствующий примеру 4.1:

SELECT Kod1, UNITPRICE * (1 + 0,18) AS ЦЕНА FROM Товары

Результат выполнения данного запроса представлен в таблице 4.2. Таблица

4.2 – Результат выборки из таблицы **Товары**

Kod1	ЦЕНА
1180.0000	
2360.0000	
11800.0000	

0 запросе после ключевого слова **AS** записано новое название столбца

таблицы.

Применение конструкции **SELECT** в формировании запросов с вычисляемыми полями позволяет использовать в них не только арифметические выражения, но и простые имена полей. Кроме того, в результаты выборки можно добавить константы.

Если в SQL-запрос, соответствующий примеру 1, добавить строку «Цена указана с учетом НДС», то он примет вид:
SELECT Kod1, UNITPRICE * (1 + 0,18) AS ЦЕНА, «Цена указана с учетом НДС» FROM Товары

Результат выполнения данного запроса представлен в таблице 3.

Таблица 4.3 – Результат выполнения запроса

Kod1	ЦЕНА	Цена указана с учетом НДС
1	1180.0000	Цена указана с учетом НДС
2	2360.0000	Цена указана с учетом НДС
3	11800.0000	Цена указана с учетом НДС

При выборке с помощью команды **SELECT** можно использовать агрегатные функции:

- 1 **MIN(X)** – вычисляет минимальное значение из множества X;
- 2 **MAX(X)** – вычисляет максимальное значение из множества X;
- 3 **AVG(X)** – вычисляет среднее арифметическое из множества значений X;

- 4 **SUM(X)** – вычисляет сумму значений множества X; 5)

COUNT(X) – определяет число элементов множества X.

Примеры использования этих функций:

0 **SELECT COUNT(*) FROM GOODS** – создает выборку, состоящую из одной строки и одного поля, содержащего количество всех строк таблицы

GOODS (товары);

1 **SELECT MAX(Цена), MIN(Цена), AVG(Цена) FROM GOODS** – создает выборку, состоящую из одной строки и трех полей, содержащих минимальное значение цены, максимальное значение цены и её среднее значение.

5 Выборка с условием

Для задания условия выборки в SQL-запросе используется команда (ключевое слово) **WHERE**. Условие, следующее за ключевым словом **WHERE**, может включать:

арифметические операторы сравнения: **=, <>, >, <, >=, <=;**

логические операторы – **AND, OR, NOT;**

скобки, определяющие порядок вычислений.

При выполнении условия выборки числа сравниваются алгебраически: отрицательные числа считаются меньше, чем положительные, независимо от их абсолютной величины. Строки сравниваются с их представлением в коде **ANSI**. При сравнении двух строк, имеющих разные длины, предварительно

более короткая строка дополняется справа пробелами для того, чтобы строки имели одинаковую длину.

Пример 5.1. Пусть существует некоторая база данных, в которой имеется таблица **Table3**, содержащая информацию о поставщиках, покупателях, товарах. Предполагается, что покупатель может приобретать товар в кредит. Произвести из данной базы выбор всех кодов (**kod**) и фамилий (**NAME**) покупателей, которые находятся в Москве и имеют кредит(**CREDITLIMIT**) более 200 000.

помощью языка SQL запрос, соответствующий примеру 1, можно представить в виде:

```
SELECT kod, NAME
FROM Table3 WHERE
CITY="Москва"
AND CREDITLIMIT>200000
```

6 Выборка с упорядочением

Вкладка **Ordered By** в окне дизайнера запросов позволяет управлять порядком расположения записей в результирующей таблице. Для упорядочивания выделите указателем (курсором) поля, которые будут определять порядок сортировки выбранных данных, и перенесите их последовательно в список **Ordering criteria**. Для каждого выбранного поля можно с помощью переключателя (кнопки) **Order options** установить критерий упорядочивания по возрастанию (**Ascending**) или по убыванию (**Descending**).

Порядок сортировки записей результирующей таблицы определяется порядком следования полей в списке **Ordering criteria** и критерием упорядочивания отдельных полей.

Пример 6.1. Для таблицы **Table3** из примера 4.1 необходимо произвести выбор кода (**kod**), имени (**NAME**) и суммы кредита (**CREDITLIMIT**) всех покупателей, проживающих в Ставрополе, расположив их в порядке убывания.

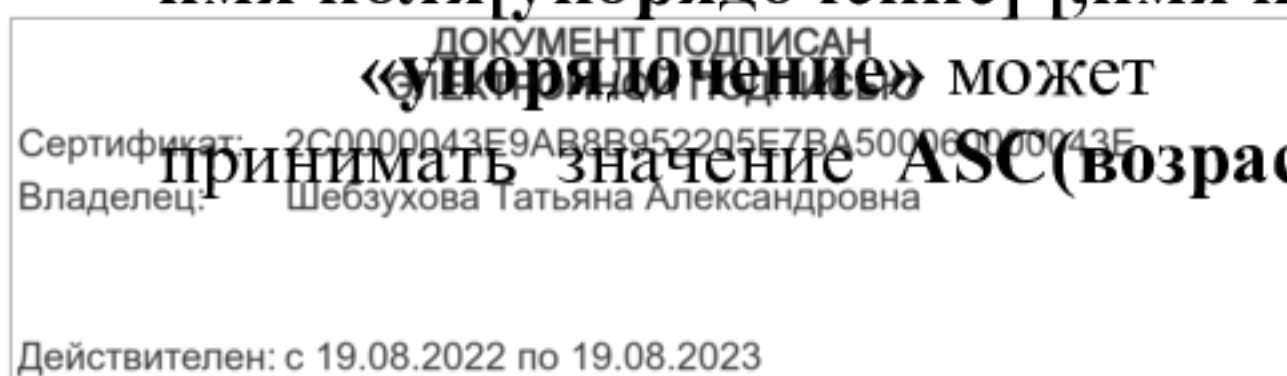
SQL-запрос, реализующий пример 6.1, имеет вид:

```
SELECT kod, NAME, CREDITLIMIT FROM
Table3 WHERE CITY="Ставрополь" ORDER BY
CREDITLIMIT DESC
```

выборках без указания критерия упорядочивания данных результирующая таблица будет упорядочена в соответствии с внутренними алгоритмами их осуществления. Из примера 6.1 следует, что результат выборки может быть организован в определенной последовательности. Упорядочение данных в выборке может осуществляться по любому полю результирующей таблицы:

имя поля[упорядочение] [,имя поля [упорядочение]] ..., где аргумент

«упорядочение» может принимать значение **ASC(возрастание)** или **DESC (убывание)**. По



умолчанию устанавливается значение **ASC**. В качестве аргументов имя поля могут использоваться только поля результирующей таблицы. Поэтому недопустима следующая конструкция:

```
SELECT kod, NAME, CREDITLIMIT  
FROM Table3  
ORDER BY CITY
```

Для идентификации полей, по которым осуществляется упорядочивание, можно использовать не только наименования полей результирующей таблицы, но и их номера (номер поля указывает порядковую позицию данного поля в результирующей таблице запроса). Благодаря этому можно упорядочить результат на основе вычисляемых полей, которые не обладают именами. Например, результатом выполнения запроса:

```
SELECT Kod1, UNITPRICE * (1 + 0,18)  
FROM Товары
```

ORDER BY 2 будет являться результирующая таблица, во второй столбец которой будет помещена информация о стоимости товаров с учетом НДС. Записи в выборке будут упорядочены по второму столбцу.

7 Выборка с использованием оператора **BETWEEN**

Для организации выборки информации из базы данных, принадлежащей некоторому диапазону, в **SQL**-запросах используются операторы:

«>(Больше)»; «<(Меньше)» и **Between (Между)**. Выбор каждого из этих операторов осуществляется в окне мастера запросов из списка **Criteria (Критерий)**, расположенного во вкладке **Filter (Фильтр)**. Операторы «>(Больше)» и «<(Меньше)» используются в том случае, если задана только нижняя или верхняя граница диапазона, а оператор **Between (Между)** – если известны обе границы диапазона.

помощью **SQL**-запроса, набранного в командном окне,

```
SELECT kod, NAME,  
UNITPRICE FROM Table4  
WHERE UNITPRICE BETWEEN 20000 AND 100000
```

на экран будет выведена таблица, содержащая сведения о товарах, стоимость которых (**UNITPRICE**) находится в диапазоне от 20000 до 100000 включительно. Оператор **BETWEEN** возвращает истинные значения и для граничных точек.

Кроме того, при организации выборки из таблицы значений, не принадлежащих некоторому диапазону, может быть использовано условие **NOT BETWEEN (не принадлежит диапазону между величинами)**, например:

```
SELECT kod, NAME,  
UNITPRICE FROM Table4  
WHERE UNITPRICE NOT BETWEEN 20000 AND 100000
```

8 Выборка с использованием оператора IN (принадлежит)

СУБД **Visual FoxPro** позволяет при создании запросов формировать несколько условий. В том случае, если все задаваемые условия накладываются на одно поле, их можно разместить в одной строке. Для этих целей используется оператор **IN (принадлежит)**.

В частности, результатом выполнения SQL-запроса, приведенного ниже

SELECT kod, NAME, UNITPRICE
FROM Table4

WHERE UNITPRICE IN (100000, 200000, 500000) является таблица, содержащая информацию о товарах, цена которых равна 100000, 200000 или 500000.

Оператор **IN** является краткой записью условия, представляющего собой последовательность отдельных сравнений, соединенных между операторами **OR (или)**. Предыдущая конструкция **SELECT** эквивалентна следующей конструкции:

SELECT kod, NAME, UNITPRICE
FROM Table4
WHERE UNITPRICE=100000
OR UNITPRICE=200000
OR UNITPRICE=500000

Как и в случае с оператором **BETWEEN** можно также использовать конструкцию **NOT IN (не принадлежит)**, например:

SELECT kod, NAME, UNITPRICE
FROM Table4
WHERE UNITPRICE NOT IN (100000, 200000, 500000)

9 Выборка с использованием шаблонов

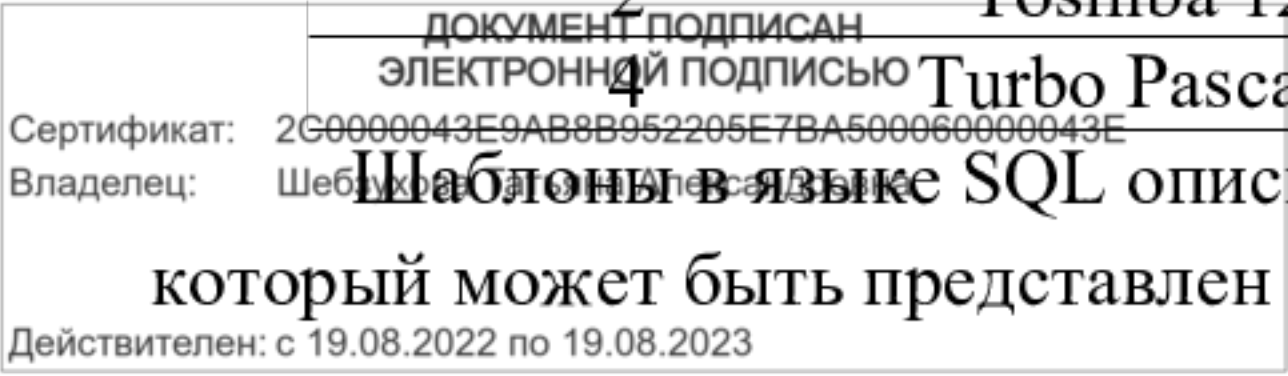
Использование шаблонов языка SQL расширяет возможности выборки информации из базы данных, для которой пользователь не помнит точные значения полей или названия полей длинные и пользователь хочет сократить время их набора. Например, с помощью запроса

SELECT kod, NAME, UNITPRICE
FROM Table4
WHERE NAME LIKE "T%"

пользователь может выбрать из базы данных все товары, наименование которых начинается с буквы "T"(таблица 9.1) . Таблица 9.1 – Результат выборки по шаблону

kod	NAME	UNITPRICE
2	Toshiba 1200	600.00
4	Turbo Pascal	200.00

Шаблоны в языке SQL описываются с помощью оператора **LIKE**, который может быть представлен в виде:



имя поля **LIKE** строковая константа

Результат выполнения оператора **LIKE** принимает значение «истина», если значение в указанном поле соответствует образцу, указанному аргументом «строковая константа», символы которой интерпретируются следующим образом:

- 0** _ (пробел или подчеркивание) – любой одиночный символ;
- 1** % (процент) – произвольная последовательность символов.
- 2** Все другие символы обозначают сами себя. Специальный смысл специальных символов отменяет знак «\».

Например, совокупность символов «\%» означает процент, а не последовательность символов.

В приведенном примере конструкция **SELECT** будет осуществлять выборку записей из таблицы **Table4**, для которых значение в поле **NAME** начинается с буквы «Т» и содержит далее любую последовательность символов. Например, если из таблицы **TAB7** необходимо выбрать все строки, которых поле **string1** содержит «+», а предпоследняя буква «S», то SQLзапрос, реализующий данное условие примет вид:

SELECT * FROM TAB7 WHERE string1 LIKE «%+%S_».

10 Выборка из связанных таблиц

Способность «соединять» две или более таблицы в одну представляет собой одну из наиболее мощных возможностей реляционных баз данных.

0 СУБД Visual FoxPro при создании многотабличного запроса в окно конструктора запросов добавляются все участвующие в выборке таблицы и определяются условия их объединения. Таблицу в окно конструктора запросов при организации выборки из нескольких связанных таблиц можно добавить одним из способов:

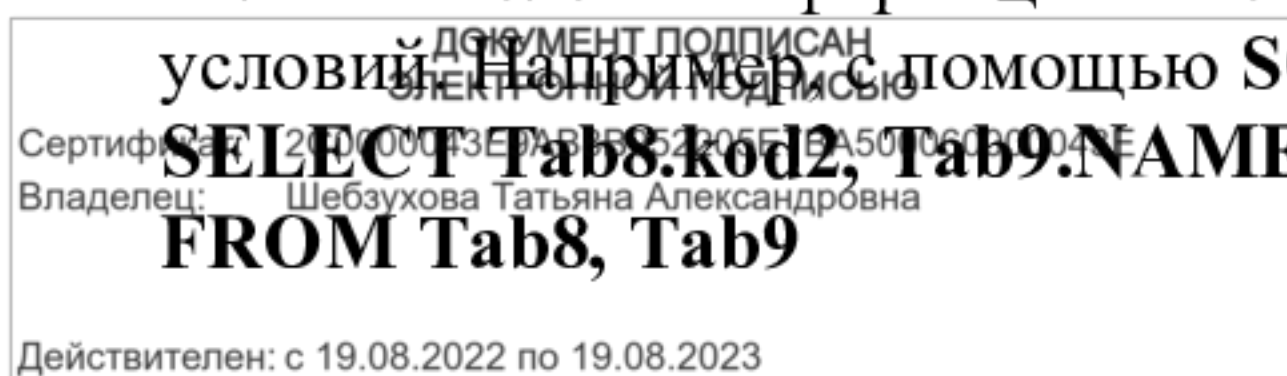
- 1** выполните команду **F10 → Query → Add Table**; 2) нажмите кнопку **Add Table**.

После этого на экран будет выведено диалоговое окно **Add Table or View**. В этом окне следует выбрать необходимые для формирования запроса таблицы, затем нажать кнопку **Add**. Таким образом, выбранные таблицы будут размещены в окне конструктора запросов. Если между участвующими запросе таблицами в базе данных установлены постоянные отношения, то в окне конструктора запросов эта связь будет отображаться в виде линии, соединяющей таблицы, а на вкладке **Join** появится запись, содержащая условие объединения таблиц.

Простое соединение

Простое соединение как один способов выбора информации из нескольких связанных таблиц предполагает, что на выбираемую пользователем информацию не накладывается никаких дополнительных условий. Например, с помощью SQL-запроса

**SELECT Tab8.kod2, Tab9.NAME
FROM Tab8, Tab9**



WHERE Tab8.STOCK= Tab9.STOCK будет получен список кодов (kod2) и наименований (NAME) проданных покупателям товаров, представленный в таблице 10.1.

Таблица 10.1 – Результат выборки из двух таблиц

Kod2	NAME
2	BORLAND C++
10	PARADOX for Windows
18	Книга TURBO C++
20	Книга C#

данном случае использованы уточнённые имена полей, т.е. ссылки на поля после ключевого слова **WHERE** уточнены именами содержащих их таблиц. Два поля **STOCK** являются ключами, с помощью которых установлены постоянные отношения между таблицами.

Соединение с дополнительным условием

При выполнении выборки информации из базы данных можно использовать механизм соединения с дополнительным условием, которое ограничивает выбираемую пользователем информацию. Использование выборки на основе соединения с дополнительным условием возможно только в том случае, если таблицы, включенные в запрос, связаны по одному и тому же ключевому полю. Например, для выбора из базы данных кодов и фамилий покупателей, которым проданы компьютеры Lenovo, служит SQL-запрос:

**SELECT ORDSALE.CUSTOMERNO,
ORDSALE.FIRSTNAME FROM ORDSALE, GOODS
WHERE ORDSALE.STOCK=GOODS.STOCK
AND GOODS.NAME=“ Lenovo”**

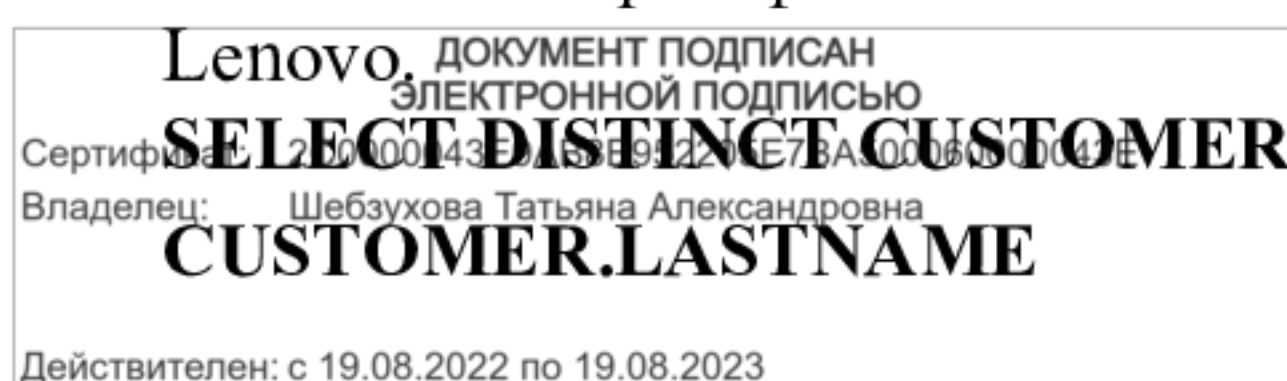
Результат выполнения данного запроса представлен в таблице 10.2. Таблица 10.2 – Результат выборки из двух таблиц с условием

ORDSALE.CUSTOMERNO	FIRSTNAME
2	Иванов
2	Иванов
11	Петров
16	Сидоров
19	Кузнецов

Соединение трех таблиц

Для выборки данных из трех таблиц в программном коде после ключевого слова **WHERE** необходимо указать два условия связи таблиц. Пример SQL-запроса для выбора информации из трех связанных таблиц: вывести на экран фамилии и имена всех покупателей, которые приобрели

**Lenovo, SELECT DISTINCT CUSTOMER.FIRSTNAME,
CUSTOMER.LASTNAME**



**FROM CUSTOMER, ORDSALE, GOODS
WHERE CUSTOMER.CUSTOMERNO=ORDSALE.CUSTOMERNO
AND ORDSALE.STOCK=GOODS.STOCK AND GOODS.NAME=“
Lenovo”**

После выполнения данного запроса на экран будет выведена таблица 10.3.

Во многих случаях выборки из нескольких таблиц используются не только для ограничения выборки, но и для объединения данных из нескольких таблиц.

Таблица 10.3 – Результат выборки из трех таблиц

CUSTOMER.FIRSTNAME	CUSTOMER.LASTNAME
Иванов	Андрей
Петрова	Дарья
Сидоров	Максим

11 Использование группировки данных при организации запросов

Группировка данных при организации запросов используется для объединения нескольких строк, включаемых в запрос, в одну.

Для выполнения группировки данных в **SQL**-запросах используется оператор **GROUP BY**. Оператор **GROUP BY** перекомпоновывает данные, включаемые в результирующую таблицу. Указание на группировку данных в **SQL**-запросе осуществляется после ключевого слова **FROM**. Сам процесс группировки данных при выполнении выборки информации из базы данных разделы или группы заключается в объединении в одну группу всех строк, которые имеют одно и то же значение поле, указанное после оператора **GROUP BY**.

Далее, к каждой группе применяется конструкция **SELECT**. Каждое из выражений, включаемое в конструкцию оператора **SELECT**, должно принимать единственное значение для группы. Это выражение может быть либо самим полем, указанным в операторе **GROUP BY**, либо арифметическим выражением, включающим это поле, либо константой, либо такой функцией как **SUM**, которая оперирует всеми значениями данного поля в группе и сводит эти значения к единственному значению.

Строки таблицы можно группировать по любой комбинации ее полей. Если поле, по значению которого осуществляется группировка, содержит какие-либо неопределенные значения, то каждое из них порождает отдельную группу.

Использование конструкции **GROUP BY** не предполагает одновременного применения в **SQL**-запросах и оператора **ORDER BY**. Для упорядочения результата, полученного после выполнения **SQL**-запроса, конструкцию **ORDER BY** <имя поля> необходимо разместить после оператора, указывающего на группировку данных.

Пусть требуется вычислить общий объем покупок для каждого товара, е. для каждого товара необходимо определить код этого товара и общий объем покупок. SQL-запрос, соответствующий данному условию, можно представить в виде:

```
SELECT kod, SUM(QUANT)
FROM ORDSALE
GROUP BY kod
```

Использование группировки данных совместно с условием

Использование группировки при выборе информации из базы данных выводит на экран все поля, значения которых совпадают. Однако на практике эта особенность группировки затрудняет работу пользователя с базой данных, так как после выполнения запроса выводятся лишние записи, значения которых в данный момент не нужны. Для ограничения выборки с упорядочиванием в этом случае используется оператор **WHERE**.

Например, SQL-запрос

```
SELECT kod, SUM(QUANT)
FROM ORDSALE
WHERE CUSTOMERNO <> 23
```

GROUP BY kod служит для выборки информации о товаре, проданном покупателям кроме покупателя с кодом 23.

Строки, не удовлетворяющие условию **WHERE**, исключаются перед группировкой данных.

Конструкции **GROUP BY** свойственно ограничение, которое заключается в том, что она работает только на одном уровне. Невозможно разбить каждую из групп на группы более низкого уровня, а затем применить некоторую стандартную функцию, например, **SUM** или **AVERAGE** на каждом уровне группировки.

Использование HAVING

Оператор **HAVING** используется для ограничения записей, участвующих в группировке, его нельзя использовать отдельно от конструкции **GROUP BY**. Оператор **HAVING** используется для того, чтобы исключать группы так же, как **WHERE** используется для исключения записей. Выражение после конструкции **HAVING** должно принимать единственное для группы значение. В частности, результатом выполнения запроса

```
SELECT kod FROM ORDSALE GROUP BY kod HAVING COUNT(*)>1
```

является таблица, в которую включены коды товаров, приобретенных более чем одним покупателем.

12 Использование квантора существования в запросах

Квантор существования является понятием, заимствованным из формальной логики. Его смысл заключается в следующем: при задании квантора существования говорят о том, что существует некоторая

переменная. В языке SQL квантор существования заменяется оператором **EXISTS (существует)**.

Пусть символ «X» обозначает некоторую произвольную переменную. Тогда формальной логике оператор с примененным квантором существования **EXISTS X** (предикат, зависящий от X) принимает значение «истина» тогда и только тогда, когда «предикат, зависящий от X» имеет значение «истина» при каком-либо значении переменной X. Если переменная X может быть любым целым числом, принадлежащим интервалу от 1 до 10, то предикат: **EXISTS X (X < 5)** принимает значение «истина», тогда как предикат **EXISTS X(X < 0)** принимает значение «ложь».

В языке SQL предикат с квантором существования представлен может быть представлен выражением вида:

EXISTS (SELECT * FROM ...)

Выражение считается истинным только тогда, когда результат вычисления подзапроса, представленного с помощью **SELECT * FROM ...**, является непустым множеством. Выражение истинно тогда и только тогда, когда существует какая-либо запись в таблице, указанной во фразе **FROM** подзапроса, которая удовлетворяет условию **WHERE** этого подзапроса.

Пример SQL-запроса с квантором существования для выбора фамилий покупателей, которым продан компьютер «Macintosh»:

SELECT NAME FROM tab10

WHERE EXISTS (SELECT * FROM tab11

WHERE tab10.kod= tab11.kod AND.STOCK = “Macintosh”)

Фактически любой запрос, который может быть выражен с использованием оператора **IN**, альтернативным образом может быть сформулирован с помощью **EXISTS**.

Можно сконструировать отрицание существования, используя

NOT EXISTS. С помощью SQL-запроса

SELECT NAME FROM tab10

WHERE NOT EXISTS (SELECT * FROM tab11

WHERE tab10.kod= tab11.kod AND.STOCK = “Macintosh”) можно получить информацию о покупателях, которые не купили «Macintosh».

Заклученный в скобки подзапрос, входящий в конструкцию **EXISTS** не обязательно использует конструкцию **SELECT * ...**. В конструкции **SELECT** можно также указать имя поля, т.е. использовать предложение вида: **SELECT имя_поля FROM ...**. Операторы **EXISTS** и **NOT EXISTS** всегда помещаются перед подзапросом.

13 Объединение множеств

Объединением двух множеств называется множество всех элементов, принадлежащих какому-либо одному или обоим множествам. Поскольку отношение является множеством записей, то можно построить объединение двух отношений. Результатом будет отношение, состоящее из всех строк,

входящих в какое-либо одно или в оба сразу отношения. Однако строки этих двух отношений должны быть совместимы по объединению.

языке **SQL** две таблицы совместимы по объединению и к ним может быть применен оператор объединения **UNION** тогда и только тогда, когда:

0 они имеют одинаковое число полей, например m ;

1 для всех i ($i = 1, m$) i -е поле первой таблицы и i -е поле второй таблицы имеют в точности одинаковый тип данных.

В **SQL-92** реализованы некоторые операции реляционной алгебры в явном виде с помощью операторов: **UNION**, **INTERSECT**, **EXCEPT**, **JOIN**.

По отношению к таблицам операторы объединения (**UNION**), пересечения (**INTERSECT**) и разности (**EXCEPT**) можно применять только в случае объединительной совместимости.

На рисунке 13.1 показано применение оператора **UNION** для объединения двух таблиц, полученное с помощью **СУБД MS Visual FoxPro**. В командном окне показан текст запроса:

SELECT name FROM table1 UNION Select name FROM table2

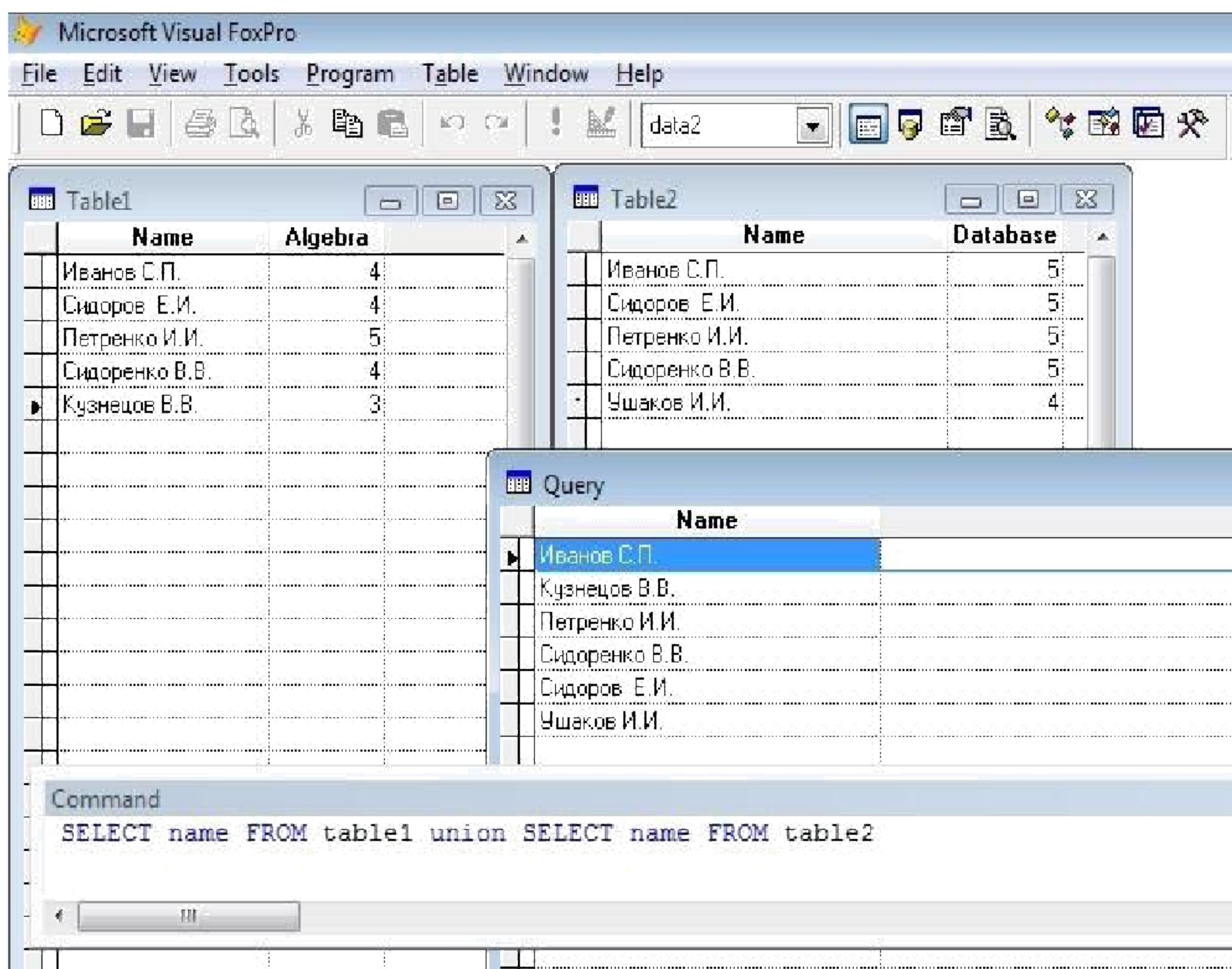
Результат выполнения запроса находится в окне Query. Состав таблиц показан в верхней части окна. В запросе нет ключевого слова **ALL**, поэтому в выборке нет повторений.

Для применения оператора **UNION** требуется совместимость таблиц по объединению. СУБД проверяет эту совместимость формально. Пользователи должны проверять и по смыслу. **НЕЛЬЗЯ ОБЪЕДИНЯТЬ ДАННЫЕ ОДНОГО ТИПА, РАЗНЫЕ ПО СМЫСЛУ!**

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

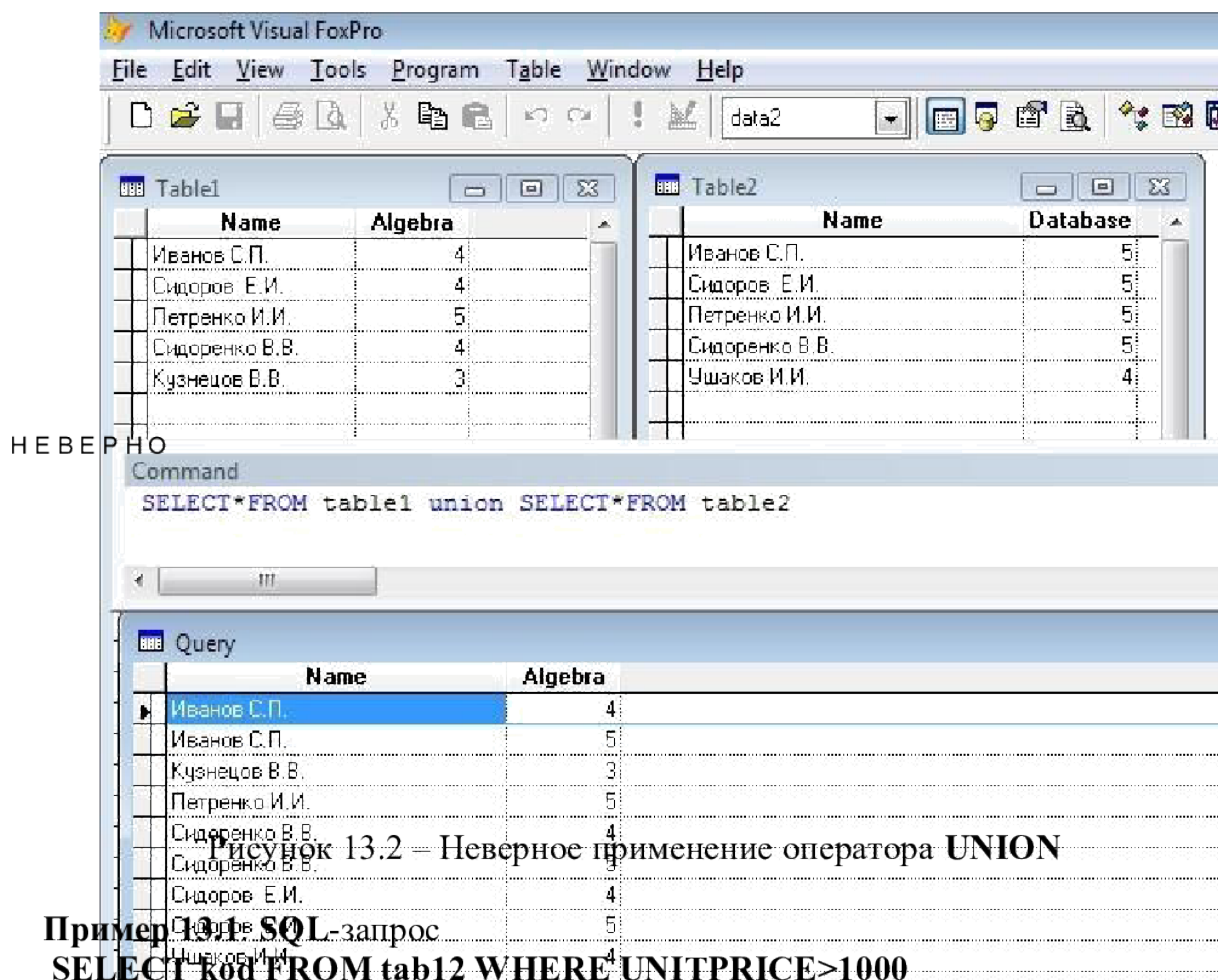
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023



унок 13.1 – Объединение таблиц, полученное средствами СУБД MS Visual FoxPro

Следующий пример показывает результат применения оператора UNION в случае, когда формальная совместимость по объединению имеется, но фактически столбцы из разных таблиц отображают разные данные. Их совмещение в едином столбце не имеет смысла. Поэтому слева от рисунка имеется надпись «неверно».



предназначен для выбора кодов товаров, которые имеют стоимость более 1000, либо приобретаются покупателем с кодом 23 (либо и то, и другое).

Использование оператора **UNION** исключает из результатов выборки повторяющиеся значения. Если это не устраивает пользователя, то вместо оператора **UNION** используют **UNION ALL**. Тогда повторы не будут исключены из выборки.

Оператором **UNION** можно соединить любое количество конструкций **SELECT**.

Пример 13.2. С помощью SQL-запроса

```
SELECT kod FROM tab12 WHERE UNITPRICE>1000
```

```
UNION
```

```
SELECT kod FROM tab13 WHERE CUSTOMERNO=23
```

```
UNION
```

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

SELECT kod FROM tab12 WHERE UNITPRICE < 500 пользователь может получить информацию о кодах товаров, которые имеют стоимость более 1000, либо приобретены покупателем с кодом 23, либо имеют цену менее 500.

Если к запросу, приведенному в примере 13.1, добавить строку **OR UNITPRICE < 500**, то его результирующая таблица будет аналогична результирующей таблице, соответствующей запросу из примера 13.2.

Оператор **ORDER BY** в запрос с использованием оператора **UNION** может входить только в последнее предложение **SELECT**. При использовании критерия упорядочивания используются номера полей результирующей таблицы. Пример с включением константы в результирующую таблицу:

```
SELECT kod, "Стоимость товара >1000$" FROM tab12  
WHERE UNITPRICE > 1000  
UNION
```

```
SELECT kod, "Товар куплен покупателем 23" FROM tab13  
WHERE CUSTOMERNO = 23 ORDER BY 2,1
```

Другой синтаксис оператора объединения:

```
(SELECT * FROM STOCK) UNION (SELECT * FROM ORDSALE)
```

Еще один вариант запроса:

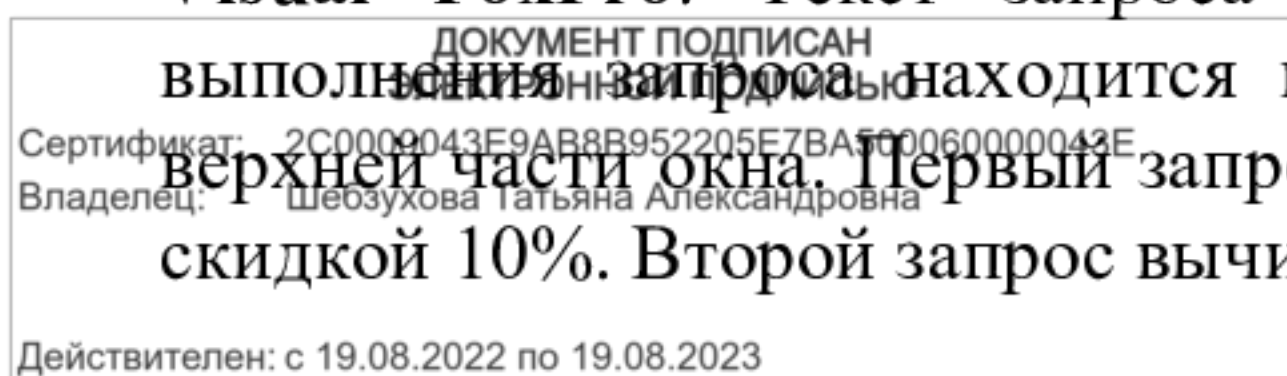
```
SELECT * FROM (TABLE tab12 UNION TABLE tab13)
```

Если эти операции необходимо применить к отдельным столбцам, то используют команду **CORRESPONDING BY (имя_поля)**. В этом случае объединительная совместимость таблиц не требуется, но обязательно указанные поля должны иметь одинаковый тип данных.

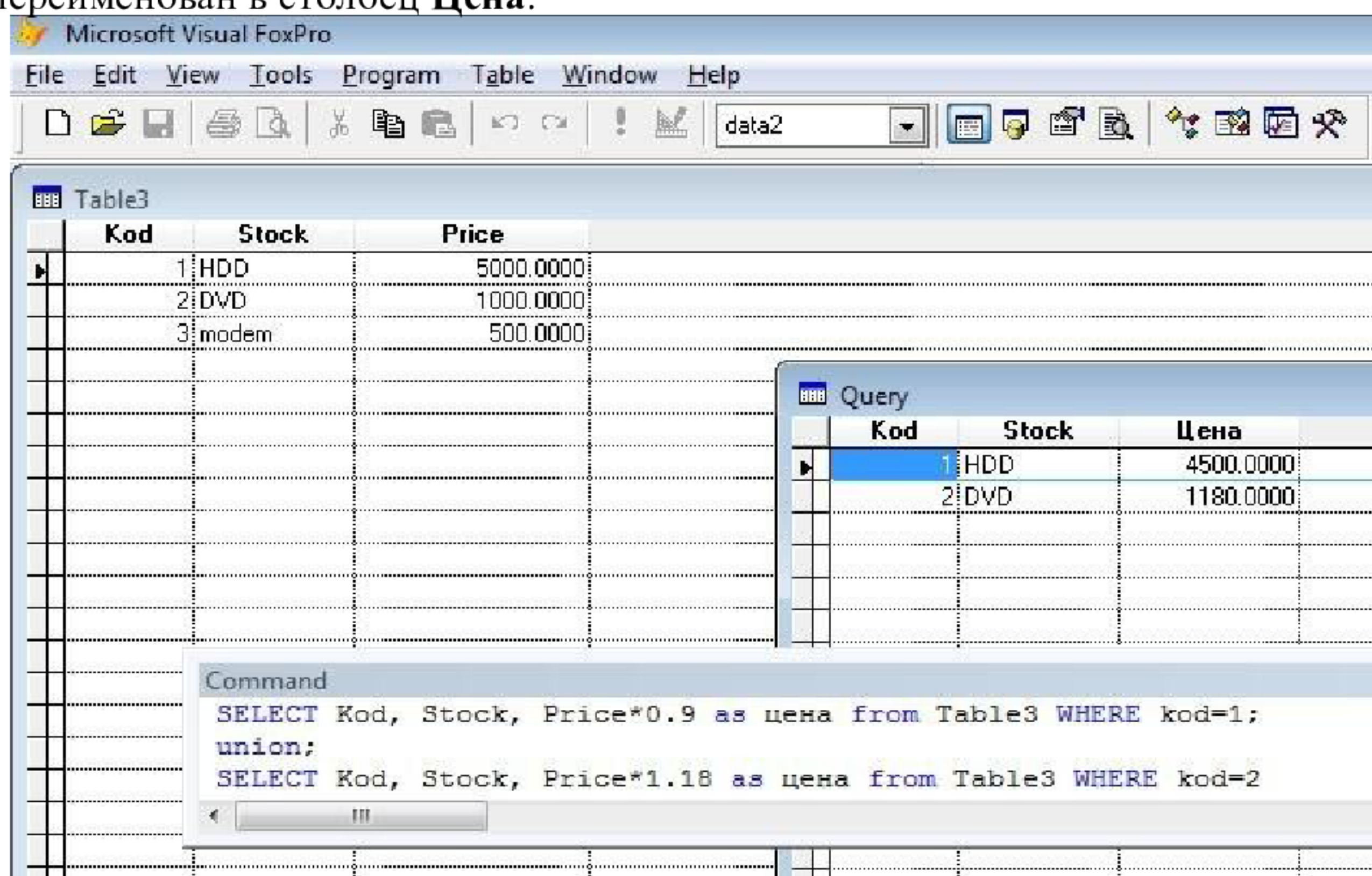
Использование оператора **UNION** при выборке вычисляемых значений

При выполнении запросов может осуществляться не только выбор из таблицы ранее введенных в нее значений, но и получение данных, отсутствующих в исходной таблице. При этом условия выборки для различных полей и строк исходной таблицы могут быть различными, что приводит к необходимости составления различных запросов. В том случае, когда полученный результат требуется представить в виде одной таблицы, требуется выполнить объединение множеств. В этом случае используется оператор **UNION**, а результаты выполнения различных запросов должны быть совместимы по объединению.

На рисунке 13.3 в СУБД **Visual FoxPro** показано применение оператора **UNION** для объединения двух запросов, полученное с помощью **СУБД MS Visual FoxPro**. Текст запроса приведен в командном окне. Результат выполнения запроса находится в окне **Query**. Состав таблицы показан в верхней части окна. Первый запрос вычисляет цену товара с кодом kod=1 со скидкой 10%. Второй запрос вычисляет цену товара с кодом kod=2 с



наценкой 18% (Это налог на добавленную стоимость). Столбец **Price** переименован в столбец **Цена**.



сунук 13.3 – Применение оператора **UNION** при выборке вычисляемых значений

Тип переменной **PRICE** – **Currency**. По умолчанию данный тип переменной настроен на получения десятичного числа с 4 знаками после запятой (рисунок 13.3, окно **Query**). Однако данный тип переменной позволяет настроить формат вывода десятичного числа таким образом, чтобы после десятичной точки было два знака, что соответствует национальной валюте России.

SQL-запрос с оператором **UNION** может объединять произвольное количество запросов. Добавим к запросу, представленному на рисунке 13.3, еще одно условие, вычислим цену товара с кодом **Kod=3** со скидкой 5%. Тогда для получения результатов выборки в виде одной таблицы потребуется SQL-запрос:

```
SELECT Kod, Stock, Price * 0.9 AS цена FROM Table3
WHERE Kod=1
UNION
SELECT Kod, Stock, Price * 1.18 AS цена FROM Table3
WHERE Kod=2
UNION
SELECT Kod, Stock, Price * 0,95 AS цена FROM Table3
WHERE Kod=3
```

ДОКУМЕНТ ПОДПИСАН
Получение вычисляемых значений и правила вычислений могут определяться условиями, которые проверяются не только в таблице, содержащей числовые данные для вычислений, но и в других таблицах.
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шабурова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Получение вычисляемых значений и правила вычислений могут определяться условиями, которые проверяются не только в таблице, содержащей числовые данные для вычислений, но и в других таблицах.

Пример 13.3. Пусть в базе данных содержится таблица **Товары(kod, price)** и таблица **Клиенты(kod_c, name, , kod)**. Чтобы вычислить цены товаров из таблицы **Товары**, одновременно предоставив клиентам из Москвы скидку 10%, а клиентам из Ставрополя скидку 5% можно использовать запрос с подзапросом:

```
SELECT Товары. kod, Товары. price*0.9 as цена, Клиенты.city  
FROM Товары, Клиенты  
WHERE EXISTS(SELECT*from Клиенты WHERE city="Москва" AND  
Товары.kod=Клиенты.kod)  
UNION
```

```
SELECT Товары. kod, Товары. price *0.95 as цена,  
Клиенты.city FROM Товары, Клиенты  
WHERE EXISTS(SELECT*from клиенты WHERE city="Ставрополь"  
AND Товары.kod=Клиенты.kod) AND Товары.kod=Клиенты.kod
```

14 Модификация данных в таблицах

0 помощью конструкций языка **SQL** можно не только производить поиск информации в базе данных, но и выполнять основные действия над таблицами:

добавлять информацию в таблицу;
модифицировать данные в таблице;
удалять информацию из таблицы.

0 языке **SQL** для модификации данных в таблицах используется конструкция **UPDATE**, которая имеет следующий синтаксис:

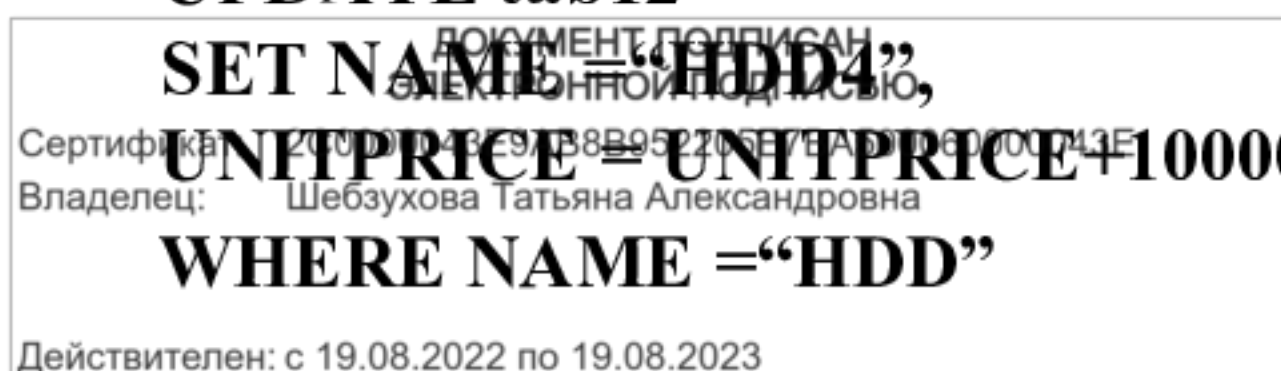
```
UPDATE таблица  
SET поле=выражение [,поле=выражение] ...  
[WHERE условие]
```

1 результате выполнения этой конструкции все записи в таблице, которые удовлетворяют условию, обновляются в соответствии с оператором присвоения «**поле = выражение**».

При использовании конструкции **UPDATE** может использоваться только одна таблица. При использовании подзапросов для модификации данных результат выборки должен возвращать только одно значение, а не несколько.

Модификация единственной записи. Для каждой записи, которая должна быть обновлена, т. е. для каждой записи, которая удовлетворяет условию **WHERE**, или для всех записей, если фраза **WHERE** опущена, ссылки во фразе **SET** на поля этой записи обозначают значения этих полей до их модификации. Например, с помощью **SQL**-запроса

```
UPDATE tab12  
SET NAME =“HDD”,  
UNITPRICE = UNITPRICE+10000.00  
WHERE NAME =“HDD”
```



будет изменено название товара «HDD» на «HDD4» и увеличена стоимость на 10000.00.

Модификация множества записей. При изменении значений во множестве записей таблицы условие, определяемой при помощи оператора **WHERE**, должно удовлетворять всему множеству. Используя SQL-запрос **UPDATE tab14**

SET CREDITLIMIT= CREDITLIMIT*2

WHERE CITY=“Ставрополь”

можно для всех покупателей, проживающих в Ставрополе увеличить в 2 раза сумму кредита.

Модификация с подзапросом. Модификация с подзапросом используется в том случае, если необходимо произвести изменения полей в связанных таблицах. В частности, с помощью SQL-запроса

UPDATE tab14

SET UNITPRICE = 0.8*UNITPRICE

WHERE “Мичуринск” = (SELECT CITY FROM tab15

WHERE tab15.CUSTOMERNO= tab14.CUSTOMERNO) можно вывести на экран список всех покупателей, проживающих в Мичуринске, уменьшив стоимость, приобретенного ими товара на 20%.

15 Удаление данных

Для удаления данных из базы в языке SQL используется конструкция **DELETE**, которая имеет следующий синтаксис:

DELETE FROM таблица [WHERE условие]

1 результате выполнения конструкции удаляются все записи, которые удовлетворяют условию.

помощью конструкции **DELETE** из базы данных может быть удалена одна запись, множество записей, все записи из одной таблицы; одна или множество записей из нескольких связанных таблиц.

Пример 15.1 – SQL-запрос, иллюстрирующий удаление одной записи из базы данных

результате его выполнения будет удален из базы данных покупатель со значением кода, равным «23».

Пример 15.2 – SQL-запрос, иллюстрирующий удаление из базы множества записей

DELETE FROM tab12 WHERE STOCK=34

После выполнения данного SQL-запроса из таблицы **tab12** будут удалены все записи, в которых поле **STOCK** равно 34.

Пример 15.3 – SQL-запрос, иллюстрирующий удаление всех записей из таблицы ORDSALE:

DELETE FROM ORDSALE

Сертификат: 201000008E4E383951205E75E267000000000043E
Владелец: Шебухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

результате выполнения этой операции таблица будет доступна для дальнейшей работы, однако, она будет пустой. Удаление из таблицы всех записей не приводит к уничтожению таблицы.

SQL-запрос с подзапросом используют в том случае, если необходимо произвести удаление записей из нескольких связанных таблиц, принадлежащих одной базе данных.

Пример 15.4 – Удаление с подзапросом **DELETE FROM tab12 WHERE “Киев”= (SELECT CITY FROM tab13 WHERE tab13.CUSTOMERNO= tab12.CUSTOMERNO)**

После выполнения данного запроса из базы данных будет удалена информация о покупках всех клиентов, проживающих в Киеве.

16 Добавление записей

Для добавления записей в базу данных с помощью языка **SQL** используется конструкция **INSERT**, которая имеет два варианта синтаксиса:

INSERT INTO таблица [(поле [,поле] ...)] VALUES

(константа [,константа]...) или

INSERT INTO таблица [(поле [,поле] ...)] подзапрос

В первом варианте в таблицу вставляется запись, имеющая заданные значения для указанных полей, причем *i*-я константа в списке констант соответствует *i*-у полю в списке полей. Во втором варианте формируется подзапрос, представляющий собой множество записей, которые добавляются в таблицу. При этом *i*-е поле результата подзапроса соответствует *i*-у полю в списке полей добавляемой таблицы. В обоих случаях отсутствие списка полей эквивалентно использованию всех полей таблицы.

С помощью конструкции **INSERT** в базу данных можно вставить одну запись, множество записей.

Пример 16.1 – **SQL**-запрос для вставки одной записи в базу данных

INSERT INTO tab15

(kod, NAME, UNITPRICE,

CATEGORY) VALUES

(1001,“Lenovo”,3000000,2)

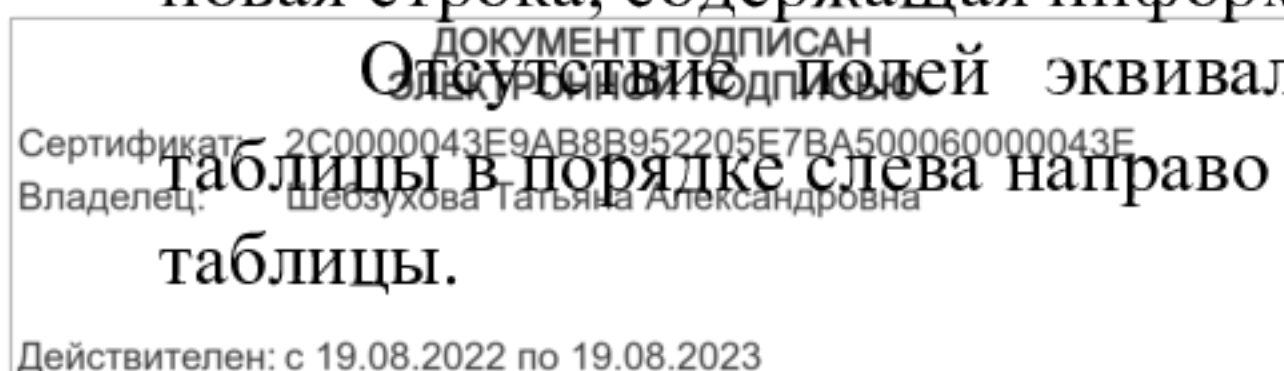
В результате выполнения приведенного выше запроса будет создана новая запись для товара с заданным номером, наименованием, стоимостью и категорией товара. В частности, в таблицу **tab15** будет добавлен товар «Lenovo», с кодом равным 1001, стоимостью 30000000 и категорией, равной 2.

Пример 16.2 – **SQL**-запрос для вставки одной записи в базу данных

INSERT INTO tab15 VALUES (1001,“Pentium”,3000000,2)

В помощью данного **SQL**-запроса в таблицу **tab15** также добавляется новая строка, содержащая информацию о товаре «Pentium».

Отсутствие полей эквивалентно перечислению списка всех полей таблицы в порядке слева направо так, как они были определены при создании таблицы.



17 Запросы по дате

При работе с базой данных у пользователя часто возникает проблема поиска информации за определенный промежуток времени. Для выполнения в СУБД **Visual FoxPro** запросов по дате можно воспользоваться следующими способами:

с помощью форм;

с помощью конструктора запросов; 3) с помощью «мастера запросов». Чтобы создать запрос с помощью конструктора запросов необходимо активизировать вкладку **Filter (Фильтр)** и выбрать из списка **Field Name (Имя поля)** поле, содержащее дату. Затем в списке **Criteria (Критерий)** выбрать значение **Between (Между)**.

После перехода на поле «**Example (Образец)**» можно вводить начальную и конечную даты интервала. Для ввода дат используется следующий формат: **CTOD(12.12.01)**, **CTOD(12.12.04)**.

Для поиска по дате можно использовать экранную форму, приведенную на рисунке 17.1.

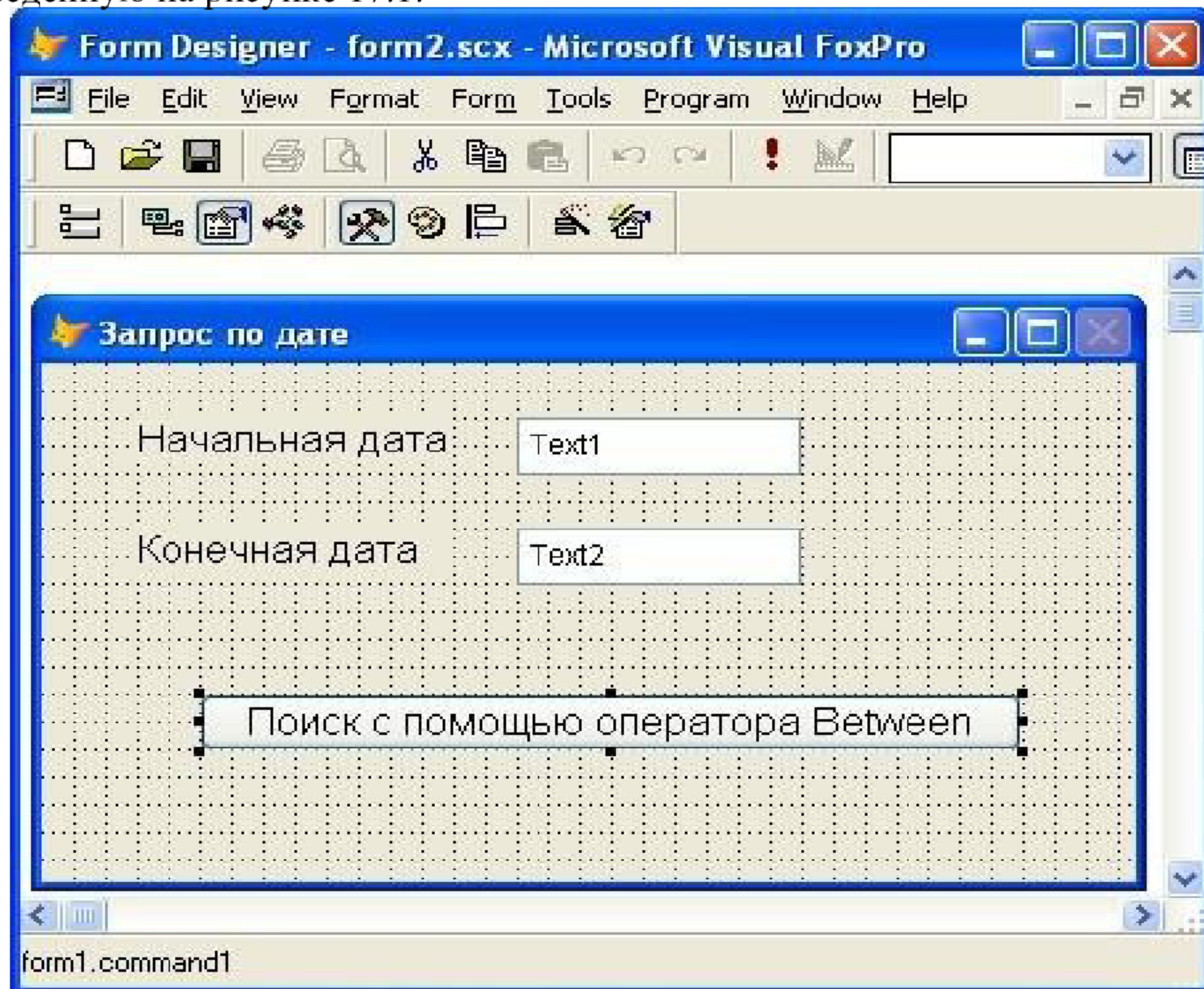


Рисунок 17.1 – Экранная форма для поиска информации по дате

качестве среды окружения для спроектированной формы нужно выбрать таблицу, содержащую данные типа «Date». Например, таблицу Table1, представленную на рисунке 17.2.

Код_товара	Наименование	Дата_поставки	Код_поставки	Количество	Цена	Сумма
1	самописец Н3095	11.04.03	1	3	14934	14934
2	тахометр ТЭ-2	13.05.04	1	1	8000	8000
3	амперметр Э 536	13.05.03	1	2	10080	20160
4	вольтметр ВЗ-59	12.06.01	1	2	28200	56400
5	тестер ДСТ-2М	08.08.02	2	1	19710	19710
6	дефектоскоп УД2-12	07.06.04	3	1	67240	67240
7	самописец Н3095	12.12.04	4	1	15000	15000
8	тахометр ТЭ-2	15.12.04	5	2	8000	16000
9	амперметр Э 536	15.12.04	5	2	10080	20160
10	вольтметр ВЗ-59	15.12.04	5	1	28200	28200
11	тестер ДСТ-2М	15.12.04	5	2	19710	35420
12	дефектоскоп УД2-12	15.12.04	5	1	67240	67240

Рисунок 17.2 – Экранная форма **Table1**

Для организации поиска информации по дате в командном окне кнопки «Поиск с помощью оператора **Between**» для процедуры **Click** введите программный код, показанный на рисунке 17.3. В нем функция **Alltrim()** удаляет все пробелы из строки, функция **CTOD()** преобразует тип переменной в переменную типа **DATE**.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

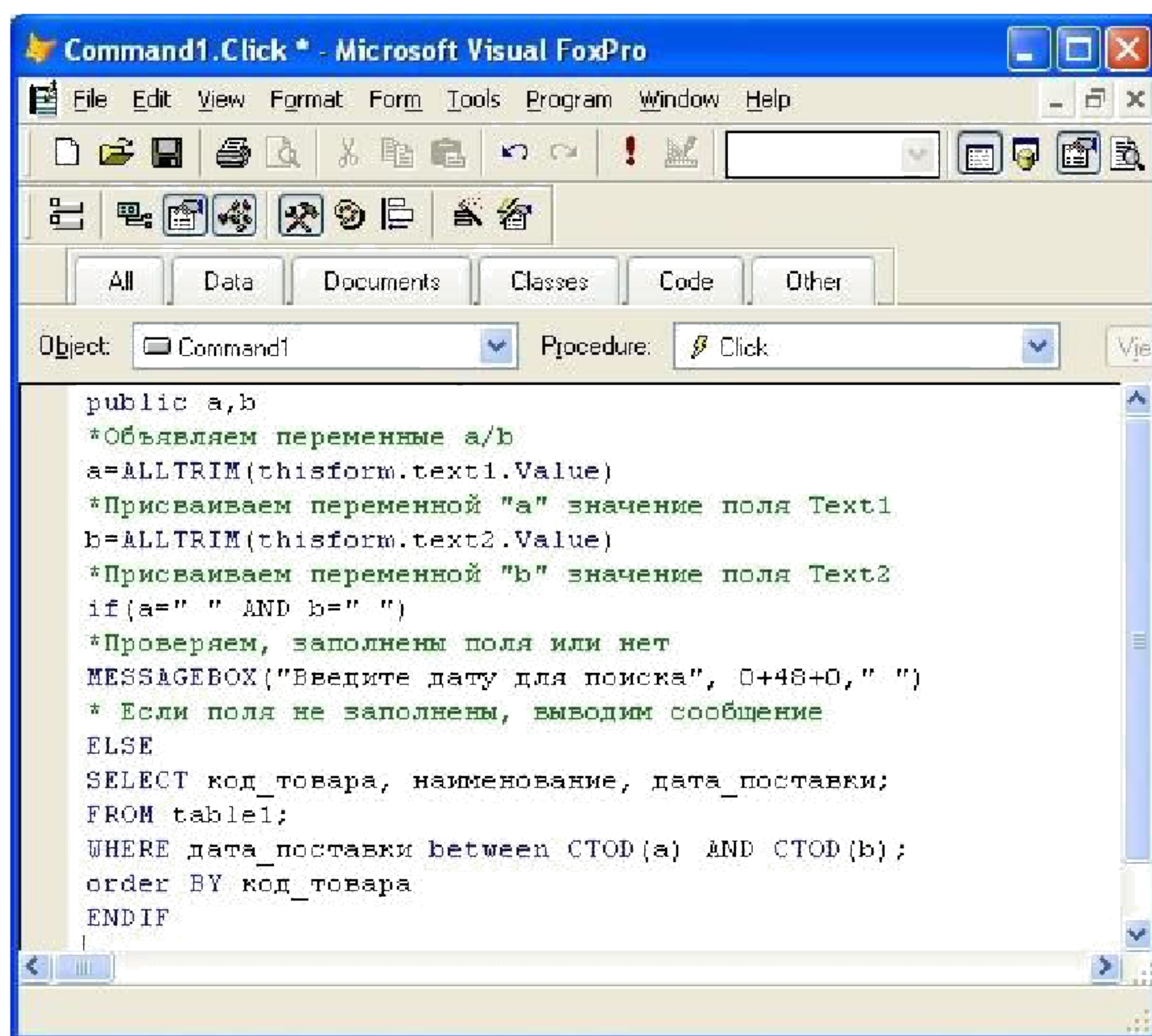


Рисунок 17.3 – Программный код для кнопки «Поиск с помощью оператора **Between**»

18. Соединение отношений. Оператор JOIN

Соединение отношений R_1 и R_2 :

$$R_1 \bowtie_{R_1(i) \text{ op } R_2(j)} R_2 \quad (R_1 * R_2),$$

обозначается символом « $\bowtie$ », под которым на месте нижнего индекса записывают условие соединения, в котором — арифметический оператор сравнения ($<$, $=$, $>$, $\neq$, $\leq$, $\geq$); k —арность отношения R_1 ; i и j —номера столбцов соответственно в отношениях R_1 и R_2 .

Если является арифметическим оператором равенства, то операцию называют эквисоединением.

Пример 18.1. Рассмотрим пример соединения отношений с поименованными столбцами $R_{11}(A,B,C)$ и $R_{12}(D,E)$. Вычислим эквисоединение при условии, что элементы столбцов B и D равны.

$$R_{13} = R_{11} \bowtie_{B=D} R_{12} = R_{11} * R_{12},$$

Отношения $R_{11}(A,B,C)$ и $R_{12}(D,E)$ представлены в таблицах 18.1 и 18. 2.

Таблица 18.1 - $R_{11}(A,B,C)$

A	B	C
a	b	c

Таблица 18.2. - $R_{12}(D,E)$

D	E
a	и

а	и	р
д	е	ж

е	к
---	---

Промежуточное действие – вычисление декартова произведения:
 $R_{11} \times R_{12}$. Результат представим в таблице 18.3.

Таблица 18.3 - $R_{11} \times R_{12}$

A	B	C	D	E
а	б	с	а	и
а	и	р	а	и
д	е	ж	а	и
а	б	с	е	к
а	и	р	е	к
д	е	ж	е	к

Следующее действие – селекция строк при условии, что элементы столбцов B и D равны. Результат содержит одну строку, представленную в таблице 18.4.

Таблица 18.4 – R_{13}

A	B	C	D	E
д	е	ж	е	к

Естественное соединение отношений R_1 и R_2 . Эта операция применяется только тогда, когда в отношениях R_1 и R_2 имеются одинаковые столбцы. Естественное соединение – это комбинация двух отношений по общим атрибутам. Условие реализации $R_1 \cap R_2 \neq \emptyset$.

Пусть отношения R_1 и R_2 имеют соответственно схемы

$R_1(A_1, A_2, \dots, A_k, B_1, B_2, \dots, B_n)$, $R_2(A_1, A_2, \dots, A_k, C_1, C_2, \dots, C_m)$, где имена $A_1, A_2, \dots, A_k$ у обоих отношений совпадают, а остальные различаются (для упрощения совпадающие имена размещены в начале, но они, конечно, могут быть записаны в любом другом порядке). Естественное соединение равно:

$$R_1 \bowtie R_2 = \pi_{A_1, \dots, A_k, B_1, \dots, B_n, C_1, \dots, C_m} (R_1 \times R_2)$$

$R_1.A_1$ — имя столбца отношения $R_1 \times R_2$, соответствующего столбцу A_1 в отношении R_1 ; $R_2.A_2$ — имя столбца отношения $R_1 \times R_2$, соответствующего столбцу A_2 в отношении R_2 . Запись «имя_таблицы.имя_атрибута», например, $R_1.A_1$ называется «уточненное имя атрибута».

Пример 18.2. Получим естественное соединение отношений ВЕДОМОСТЬ_1(Фамилия, алгебра) и ВЕДОМОСТЬ_2(Фамилия, геометрия). Отношения представлены в таблицах 18.5 и 18.6. В этих отношениях имеется один совпадающий столбец. Условие естественного соединения – в одноименных столбцах атрибуты должны принимать совпадающие значения.

Таблица 18.5 - ВЕДОМОСТЬ_1

Таблица 18.6 - ВЕДОМОСТЬ_2

Фамилия	алгебра
Иванов	отлично
Петров	хорошо

Фамилия	геометрия
Иванов	хорошо
Петров	отлично

Получим декартово произведение отношений ВЕДОМОСТЬ_1 и ВЕДОМОСТЬ_2. Результат представим в таблице 18.7. Таблица 18. 7 – отношение ВЕДОМОСТЬ_1 * ВЕДОМОСТЬ_2

Фамилия	алгебра	Фамилия	геометрия
Иванов	отлично	Иванов	хорошо
Петров	хорошо	Иванов	хорошо
Иванов	отлично	Петров	отлично
Петров	хорошо	Петров	отлично

Следующий шаг – селекция. Выбираем строки, удовлетворяющие условию: ВЕДОМОСТЬ_1.фамилия= ВЕДОМОСТЬ_2.фамилия. Получим таблицу 18. 8.

Таблица 18. 8 – Результат селекции

ВЕДОМОСТЬ_1.фамилия= ВЕДОМОСТЬ_2.фамилия (ВЕДОМОСТЬ_1 * ВЕДОМОСТЬ_2)

Фамилия	алгебра	Фамилия	геометрия
Иванов	отлично	Иванов	хорошо
Петров	хорошо	Петров	отлично

Для получения окончательного результата в таблице 18.8 нужно устранить избыточность. Для этого выполним проекцию на все разные столбцы. Результат – в таблице 18. 9.

Таблица 18.9 – результат вычисления естественного соединения отношений ВЕДОМОСТЬ_1 и ВЕДОМОСТЬ_2

Фамилия	алгебра	геометрия
Иванов	отлично	хорошо
Петров	хорошо	отлично

Для реализации соединения в языке SQL используется оператор **JOIN**. СУБД **Visual FoxPro** также используется оператор **JOIN**. Для получения естественного соединения в **Visual FoxPro** используется оператор **INNER JOIN**.

Пример 18.3. Рассмотрим базу данных, содержащую две таблицы, представленные на рисунке 18.1. Таблица Table1 представляет собой ведомость по алгебре – **Table1(Name, Algebra)** и таблица Table.2 представляет собой ведомость БД – **Table2(Name,Database)**. Условие применения естественного соединения выполняется, так как у таблиц

имеется общий столбец – **Name**. Естественное соединение таблиц можно получить разными способами, показанными на рисунке 18.1. В командном окне представлены два различных запроса, выполнение которых дает одинаковый результат, представленный в окне **Query**. Один из запросов содержит оператор **INNER JOIN**:

SELECT Table1.name, Table1.Algebra, Table2.Database FROM Table1, Table2 WHERE Table1.name= Table2.name

Другой запрос:

SELECT Table1.name, Table1.Algebra, Table2.Database FROM Table1 INNER JOIN Table2 ON Table1.name= Table2.name

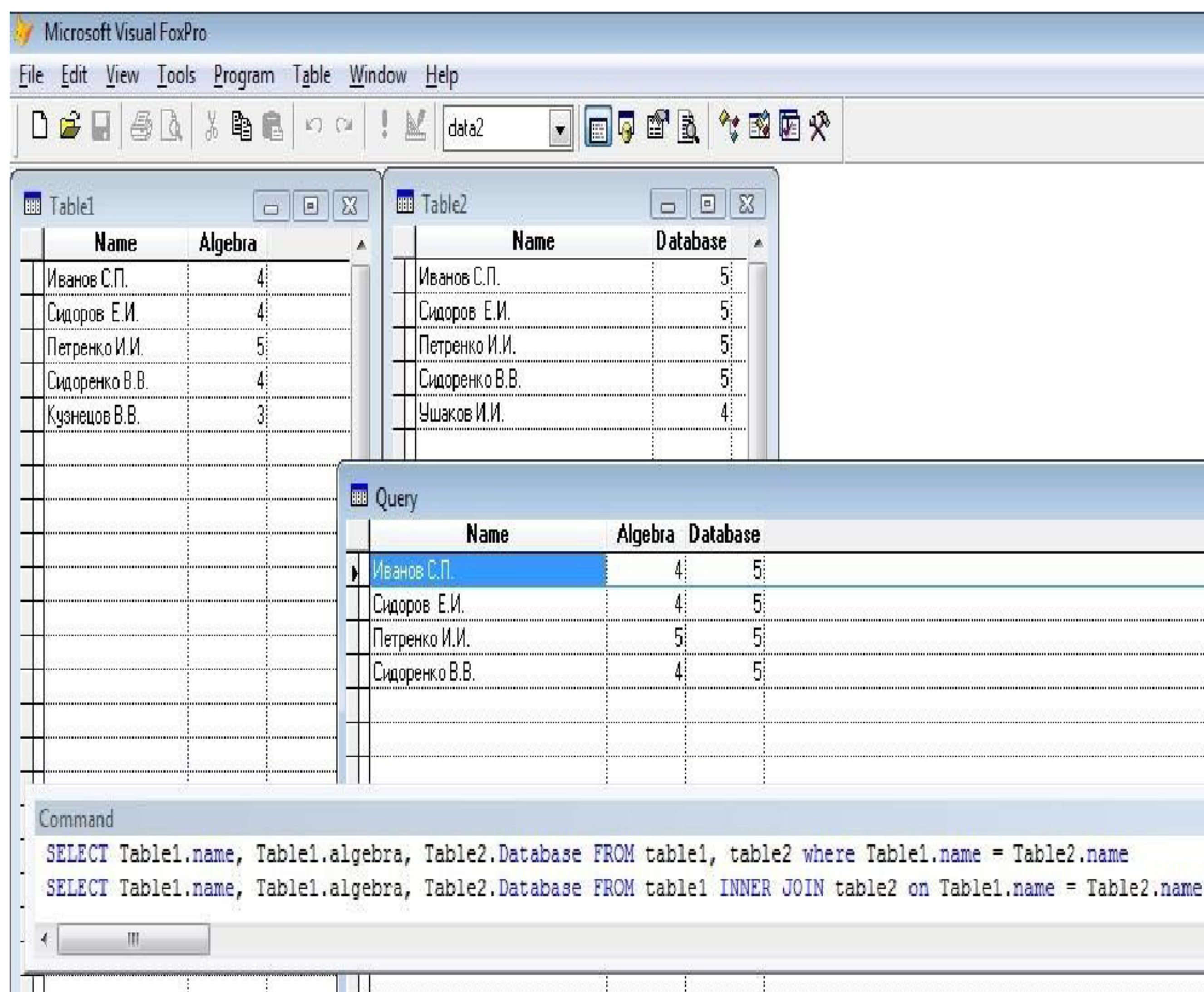


Рисунок 18.1 – Получение естественного соединения

Естественное соединение, полученное в результате выполнения любого из приведенных запросов, показано в окне Query на рисунке 18.1.

Пример 18.4. Левое соединение включает все строки левой таблицы (по отношению к оператору **LEFT JOIN**) и только те строки из правой таблицы, которые удовлетворяют условию соединения (Table1.name=Table2.name).

Запрос для получения левого соединения:

**Select Table1.name, Table1.Algebra, Table2.Database
FROM Table1 LEFT JOIN Table2 ON Table1.name= Table2.name**

Результат выполнения этого запроса представлен на рисунке 18.2. В рассмотренном примере в правой таблице Table2 нет строки, где поле Table2.name = «Кузнецов В.В.», поэтому в результате выполнения запроса появляется неопределенное значение NULL.

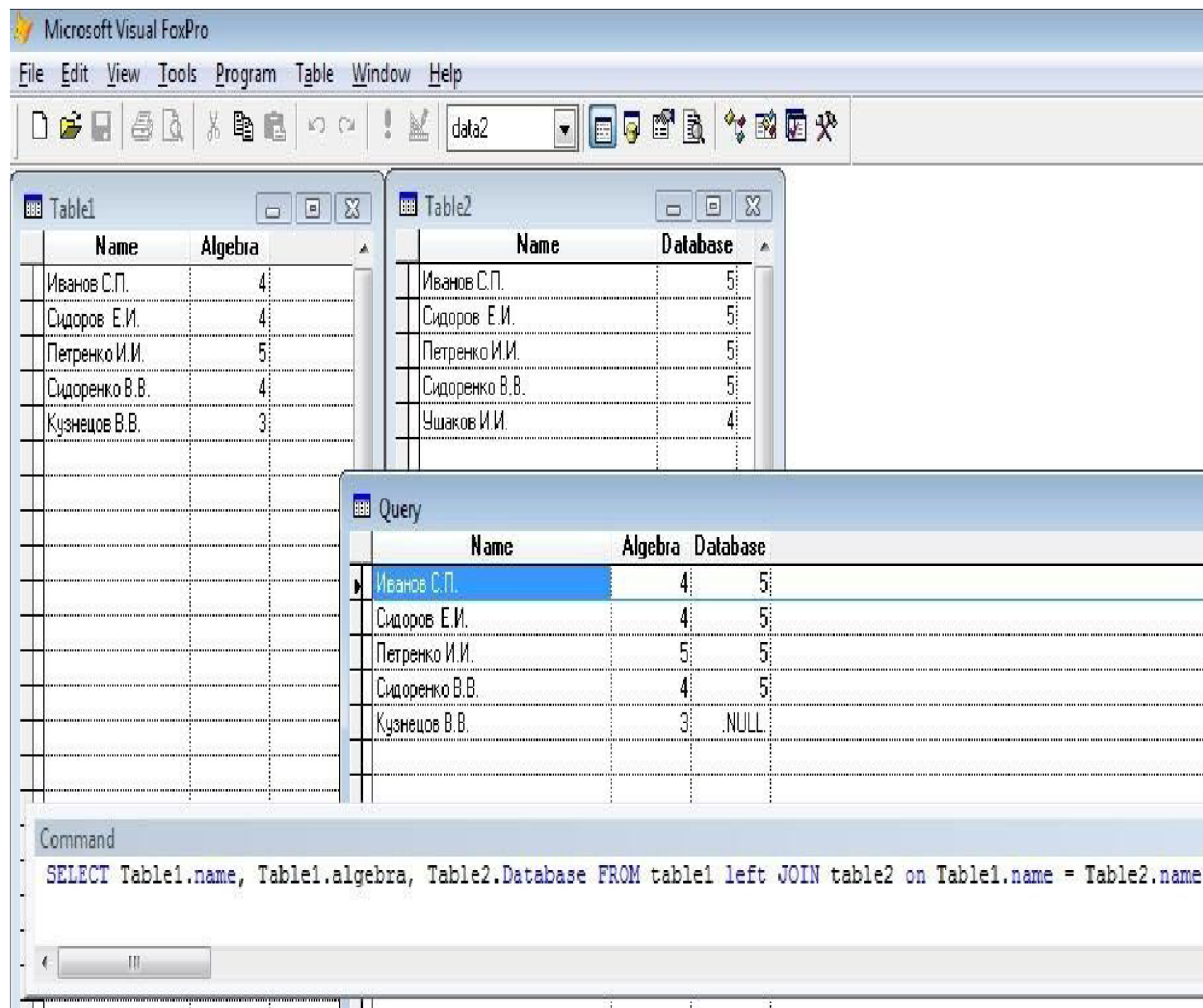


Рисунок 18.2 – Получение левого соединения

Пример18.5. Правое соединение включает все строки правой таблицы (по отношению к оператору RIGHT JOIN) и только те строки из левой таблицы, которые удовлетворяют условию соединения (Table1.name=Table2.name). Запрос для получения правого соединения

**Select Table1.name, Table1.Algebra, Table2.Database FROM Table1
RIGHT JOIN Table2 ON Table1.name= Table2.name**

Результат выполнения этого запроса представлен на рисунке 18.3.

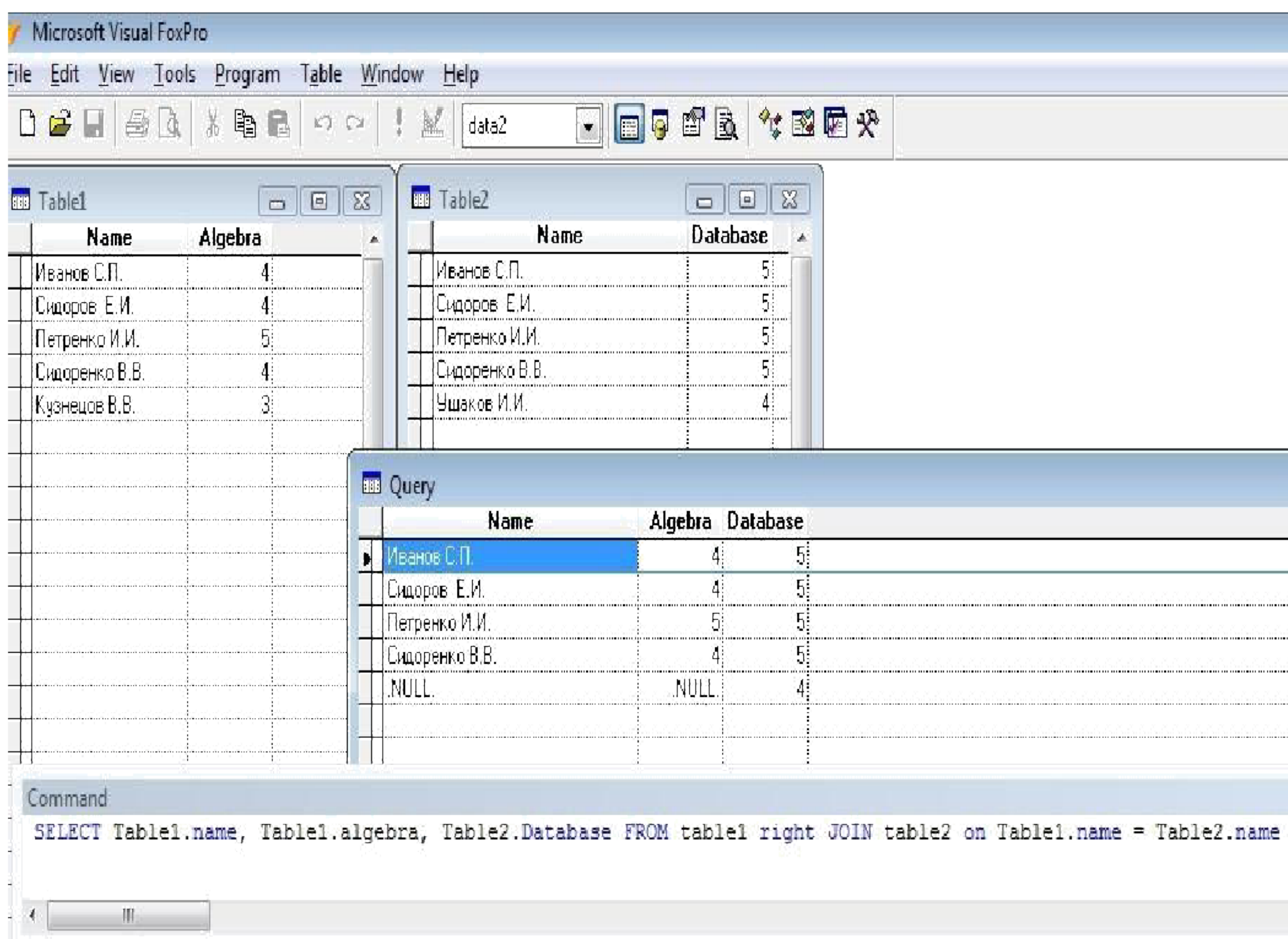


Рисунок 18.2 – Получение правого соединения

Коррелированные и некоррелированные подзапросы

В некоторых случаях необходимо выбрать данные из таблицы, основываясь на результатах дополнительных выборок из этой же таблицы. Такие выборки называются коррелированными. Некоррелированным называется подзапрос, который не зависит ни от какого внешнего запроса. Некоррелированный подзапрос не зависит от строки, которую рассматривает внешний запрос. Внешним или главным запрос называется запрос, в котором содержатся все подзапросы. Синтаксис подзапроса такой же, как и у внешнего запроса. Подзапрос записывают в скобках.

Рассмотрим рисунок 19.1, на котором представлен пример получения коррелированного подзапроса. В левой части рисунка показана таблица Table3. В командном окне записан запрос с подзапросом, который использован для выборки товаров, у которых цена меньше, чем средняя. В правой части рисунка показан результат выполнения запроса (Query).

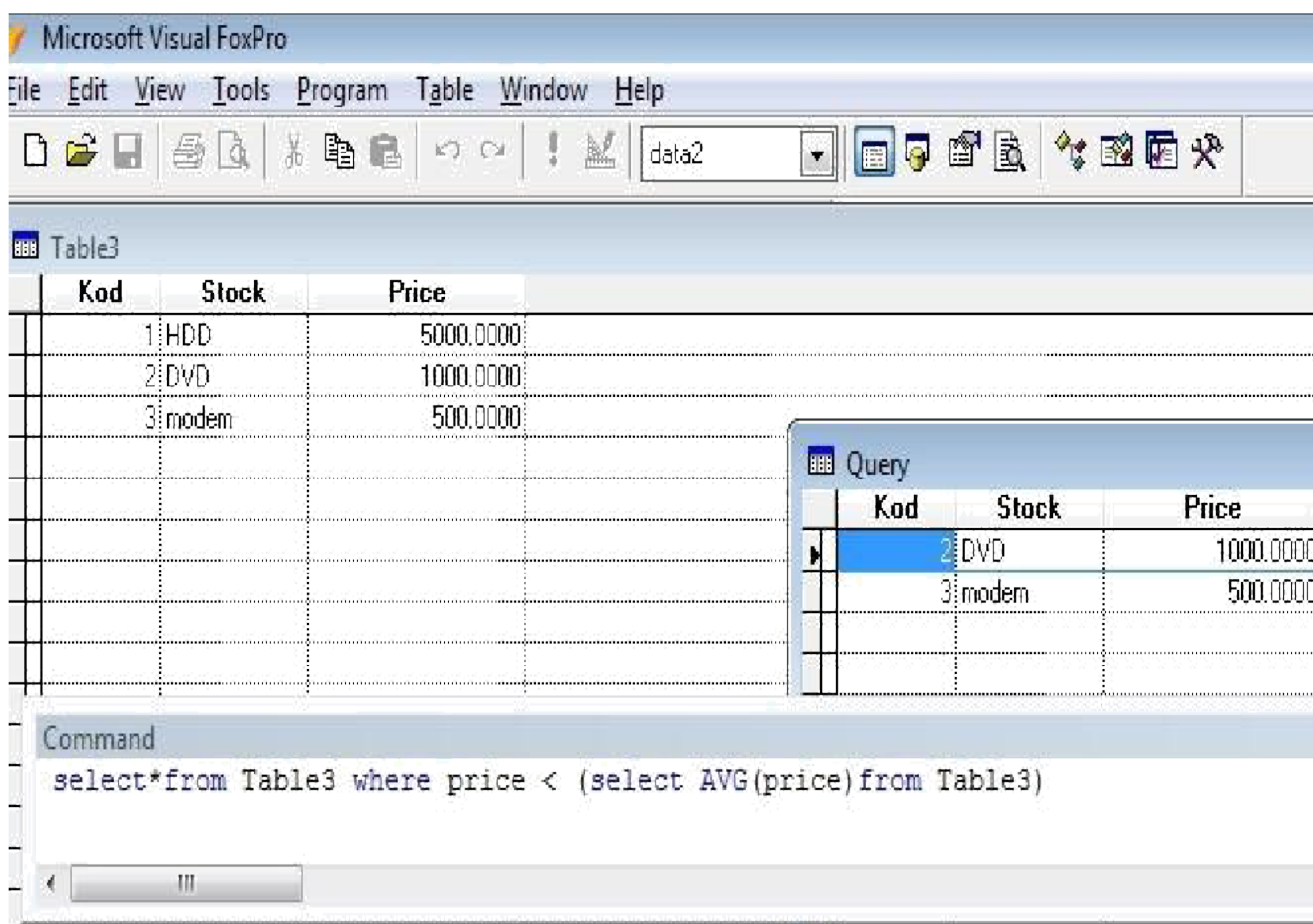


Рисунок 20.1 – Получение коррелированного подзапроса

20. Вычисления в таблицах. Команда

CALCULATE В СУБД Visual FoxPro команда

CALCULATE *eExpressionList* [*Scope*] [*FOR Expression1*]

[*WHILE Expression2*] [*TO VarList* | *TO ARRAY ArrayName*]

[*NOOPTIMIZE*] [*IN nWorkArea* | *cTableAlias*] выполняет финансовые и

статистические операции с полями таблиц или с выражениями, включающими поля, используя в качестве элементов *eExpressionList* приведенные в таблице 20.1 функции.

Таблица 21.1- Функции, употребляемые в eExpressionList

команды CALCULATE

ФУНКЦИЯ	ОПИСАНИЕ
<i>AVG(nExpression)</i>	Вычисляет среднее арифметическое всех значений <i>nExpression</i>
<i>CNT()</i> или <i>COUNT()</i>	Возвращает число записей, удовлетворяющих условиям, заданным <i>Scope</i> , <i>FOR</i> и <i>WHILE</i>
<i>MAX(eExpression)</i>	Возвращает максимальное из всех значений <i>eExpression</i> . Тип <i>eExpression</i> может быть <i>Character</i> , <i>Date</i> , <i>DateTime</i> , <i>Currency</i> и любой числовой

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

<i>MIN(eExpression)</i>	Возвращает минимальное из всех значений <i>eExpression</i> . Тип <i>eExpression</i> может быть <i>Character</i> , <i>Date</i> , <i>DateTime</i> , <i>Currency</i> и любой числовой
<i>NPV(nExpression1, nExpression2 [, инвестиции, используя ставку дисконтирования nExpression3]) nExpression1</i> , а также стоимости (<i>nExpression1</i>) будущих выплат (отрицательные значения) и поступлений (положительные значения). Ставка дисконтирования <i>nExpression1</i> задается не в процентах, а в виде числа от 0 до 1; <i>nExpression2</i> задает поле или выражение с именем поля или числовое выражение, значения которого интерпретируются как будущие выплаты или поступления. <i>nExpression1</i> - необязательное начальное вложение капитала. Не включается, если предполагается произвести начальное вложение в конце первого периода; такое начальное вложение задается в первой записи поля, причем в виде отрицательного числа	
<i>STD(nExpression)</i>	Вычисляет стандартное (средне квадратичное) отклонение величин, возвращаемых <i>nExpression</i> . Средне квадратичное отклонение <i>STD</i> величин $x_1, x_2, \dots, x_n$ от a вычисляется по формуле $\sqrt{\frac{(x_1 - a)^2 + (x_2 - a)^2 + \dots + (x_n - a)^2}{n}}$ <i>STD</i> =
<i>VAR(nExpression)</i>	Вычисляет дисперсию D величин, возвращаемых <i>nExpression</i> . Она равна квадрату стандартного отклонения: $D = STD^2$.

Помимо перечисленных в таблице функций, выражения *eExpressibnList* могут содержать и иные функции. Однако функции таблицы не могут быть аргументами иных функций. Так, можно записать команду `calculate Avg(Sqrt(Price))`, но следующая команда недопустима: `calculate Sqrt(Avg(Price))`

результатах, возвращаемых функциями, учитываются только записи, удовлетворяющие условиям, заданным *Scope*, **FOR** и **WHILE**. Поля, содержащие **NULL**, при вычислениях **CALCULATE** не учитываются.

Пример 20.1. Вычисляются средняя и максимальная цены книг, а также стандартное отклонение цен по данным, имеющимся в таблице *Books*. Результаты сохраняются в массиве *calcResults*. В последний элемент массива заносится количество вовлеченных в вычисления записей.



Записываем результаты в массив calcResults calculate Avg(Price), Max(Price), Std(Price), Cnt() to array calcResults

Печать результата

DISPLAY MEMORY LIKE

CALCRESULTS *Возможный результат:*

CALCRESULTS

N 195.71 (195.71428571)

N 210.00 (210.000000000)

N 10.33 (10.32630878) (4) N 14 (14.000000000)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Министерство науки и высшего образования Российской Федерации
Государственное образовательное учреждение высшего образования

«Северо-Кавказский федеральный университет»
Пятигорский институт (филиал) СКФУ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ по
дисциплине «Управление данными»
Направление 09.03.02 «Информационные системы и
технологии» Квалификация выпускника – бакалавр

Пятигорск, 2022 г.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

СОДЕРЖАНИЕ

<u>Введение</u>	3
<u>1. Место дисциплины в структуре ООП</u>	3
<u>2. Организационно-методические рекомендации по освоению дисциплины</u>	4
<u>3. Методические указания по выполнению самостоятельной работы студентов</u>	5
<u>4. Содержание самостоятельной работы</u>	6
<u>4.1. Примерная тематика самостоятельной работы студентов</u>	7
<u>5. План-график выполнения СРС</u>	8
<u>6. Организация контроля знаний студентов</u>	8
<u>6.1. Формы контроля знаний студентов</u>	8
<u>6.2. Рекомендации по подготовке к экзамену</u>	9
<u>7. Рекомендации по работе с литературой и источниками</u>	9
<u>7.2. Перечень рекомендуемой литературы</u>	11
<u>7.1.1. Основная литература:</u>	11
<u>7.1.2. Дополнительная литература:</u>	12
<u>7.1.3. Интернет-ресурсы:</u>	12
http://www.citforum.ru/database/	12
http://www.acm.org/sigmod/publications.html	12
http://www.acm.org/sigmod/databaseSoftware/index.html	12
<u>Материально-техническое обеспечение дисциплины</u>	12

Введение

Цели и задачи дисциплины - изучение программных, математических, технических, алгоритмических и лингвистических методов и средств, направленных на сбор, хранение, обработку и выдачу информации средствами СУБД. Задачей дисциплины "Управление данными" является не только знакомство студентов с программными средствами реализации информационных систем, но получение ими навыков практической работы с локальными базами данных. Целью курса является также рассмотрение перспектив развития СУБД, связанных интеллектуальной обработкой данных, с обработкой и хранением мультимедиадокументов, применением распределенной обработки данных, а также приобретение практических навыков обследования предметной области, концептуального, логического и физического проектирования базы данных. Задачей курса является также воспитание коммуникационной готовности студентов к работе в области информационного обмена.

ДОКУМЕНТ ПОДПИСАН
Электронно-цифровой подписью
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

1. Место дисциплины в структуре ООП

Содержание дисциплины определено ФГОС ВО по направлению подготовки "Информационные системы и технологии".

Основные понятия банков данных и знаний; информация и данные; предметная область банка данных; роль и место банков данных в информационных системах; пользователи банков данных; преимущества централизованного управления данными; база данных как информационная модель предметной области; система управления базой данных (СУБД); администратор базы данных; архитектура банка данных; инфологическое проектирование базы данных; выбор модели данных; иерархическая, сетевая и реляционная модели данных, их типы структур, основные операции и ограничения; представление структур данных в памяти ЭВМ; современные тенденции построения файловых систем; обзор промышленных СУБД; тенденции развития банков данных.

результате освоения дисциплины студент должен
знать:

методы предпроектного обследования;
методы рабочего проектирования;
основные положения теории баз данных, хранилищ данных;
методы выбора исходных данных для проектирования;
методы обеспечения безопасности и целостности данных
информационных систем и технологий;

уметь:

проводить анализ предметной области,
выявлять информационные потребности и разрабатывать
требования к информационным системам и базам данных;
формулировать запросы к базам данных на естественном
языке, с помощью абстрактных реляционных языков и на
языке SQL;
выбирать исходные данные для проектирования;
обеспечивать безопасность и целостность данных; владеть:
методами проектирования баз данных; методами
обеспечения безопасности и целостности данных; методами
предпроектного обследования;
методами выбора исходных данных для проектирования.

Организационно-методические рекомендации по освоению дисциплины

Самостоятельная работа студентов является важнейшим условием формирования научного способа познания. Она проводится накануне каждого семинарского (практического, лабораторного) занятия и включает подготовку реферата или сообщения, а также выполнение лабораторного задания.

<u>Название</u>	<u>Цель</u>	<u>Форма</u>	<u>Задания</u>	<u>Требования к</u>	<u>Рекомендуемая</u>
<u>раздела (тем)</u>		<u>контроля</u>	<u>для СРС</u>	<u>представлени</u>	<u>литература</u>
Сертификат: Владелец:	ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ 2C00000043E9AB8B952205E7BA50006C000FCE Шебзухова Татьяна Александровна	<u>СРС</u>		<u>ю и</u> <u>оформлению</u> <u>результатов</u>	
Действителен с 19.08.2022 по 19.08.2023					

				<u>СРС</u>	
Подготовка к лабораторным занятиям	изучить создание и ведение баз данных, команды SQL для создания запросов	индивидуальное собеседование	Сформулировать ответы на контрольные вопросы к лабораторным работам	Устно ответить на контрольные вопросы к лабораторным работам	-Дроздова, В. И. (СевероКавказский федеральный университет). Управление данными : учеб. пособие : Направление подготовки 320400.62- Информационные системы и технологии. Бакалавриат / В. И.Дроздова ; Сев.-Кав. федер. ун-т- Ставрополь : СКФУ, 2013. - 170 с. -Крис, Фиайли . SQL
Подготовка к лекционным занятиям	изучить основные понятия БД и СУБД	индивидуальное собеседование	сформулировать и письменно ответить на вопросы для контроля владения компетенциями данного раздела программы	письменно ответить на вопросы для контроля владения компетенциями данного раздела программы	Электронный ресурс / Фиайли Крис ; пер. А. В. Хаванов. - SQL,2019-04-19. - Саратов : Профобразование, 2017. - 452 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-44880103-7 -Швецов, В.И. Базы данных Электронный ресурс : учебное пособие / В.И. Швецов. - Базы данных,2019-12-01. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 218 с. - Книга находится в базовой версии ЭБС IPRbooks. -Агальцов, В. П. Базы данных : учебник / В. П. Агальцов, Кн.2, Распределенные и удаленные базы данных. - М. : Форум : ИНФРА-М,
Самостоятельное изучение тем "Бесфайловая организация данных" и "Оптимизация запросов"	изучить новые технологии управления данными и организацию БД	индивидуальное собеседование	сформулировать и письменно ответить на вопросы	письменно ответить на вопросы для контроля владения компетенциями	2014. - 272 с. : ил. - (Высшее образование). - Гриф: Доп. УМО. - Библиогр.: с. 260. - ISBN 978-5-8199-0394-0. - ISBN 978-5-16-003526-0 - Карпова, И. П. Базы данных : курс лекций и материалы для практических занятий :
я запросов" (Сроки выполнения – 6 семестр, 7,8 недели. Форма контроля - опрос)	средствами СУБД		для контроля владения компетенциями данного раздела	данного раздела	

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

6 семестр		
3	Раздел 3. Тенденции развития систем управления данными	10
4	Раздел 4. Организация данных	15

Организация контроля знаний студентов

6.1. Формы контроля знаний студентов

Контроль и оценка знаний, умений и навыков студентов осуществляется на лабораторных занятиях, консультациях, при сдаче экзамена. В ходе контроля знаний преподаватель оценивает понимание студентом содержания дисциплины «Управление данными», его способность анализировать развитие информационных систем и технологий.

Контроль знаний студентов может осуществляться в следующих формах:

текущий контроль знаний;
итоговый контроль знаний.

Текущий контроль знаний студентов имеет целью:
дать оценку работы каждого студента по усвоению им учебного материала, выявить недостатки в его подготовке и оказать практическую помощь в их устранении; Основными формами текущего контроля знаний студентов являются:

устный контрольный опрос; защита
лабораторной работы;
проверка конспектов лекций.

Устный контрольный опрос студентов проводится на лекциях (и лабораторных занятиях). По его результатам преподаватель оценивает качество подготовки студента к занятию.

На лабораторных занятиях знания и практические навыки студентов оцениваются по 4-балльной системе. Полученные оценки выставляются в журнале.

При проверке конспектов лекций дается анализ качества их ведения. Отмечаются допущенные ошибки, в рецензии преподавателя оценивается качество конспектирования учебного материала, даются рекомендации по улучшению качества конспектирования лекционного материала.

6.2. Рекомендации по подготовке к экзамену

Подготовка к зачету начинается с начала изучения дисциплины.

Необходимо посещать все лекции и лабораторные занятия.

Экзамен, как итоговый контроль знаний студентов имеет целью проверить и оценить учебную работу студентов, уровень полученных знаний и практических навыков.

Экзамен проводится в 8 триместре после защиты всех лабораторных работ в объеме учебной программы.

7. Рекомендации по работе с литературой и источниками

Изучение литературы и источников необходимо начинать с прочтения соответствующих глав учебных изданий, учебных пособий или литературы, рекомендованной в качестве основной или дополнительной по дисциплине «Управление данными», которые прямо или косвенно относятся к изучаемой теме.

При изучении литературы и источников студенту рекомендуется вести краткий конспект. Однако не следует переписывать все содержание изучаемой темы, нужно выписывать лишь основные идеи и главные на ваш взгляд мысли. В отдельных случаях, когда встречаются важные определения, понятия, необходимый фактический материал и примеры, статистическая информация, имеющие отношение к изучаемой теме, студенту следует выписать их в виде цитат с полным указанием библиографических источников.

Конспектирование рекомендуемой литературы и источников необходимо вести с распределением собранных материалов по отдельным главам и параграфам согласно учебно-тематическому плану. Необходимо выписывать все выходные данные по используемой литературе и источникам.

Важным этапом при работе с рекомендуемой литературой и источниками является изучение законодательных и нормативных актов федерального, регионального, местного и ведомственного уровней. При изучении Указов Президента РФ, Законов и Кодексов РФ, постановлений, положений, рекомендаций и т.д., студент должен выяснить все изменения и дополнения, которые могли быть внесены после их выхода в свет.

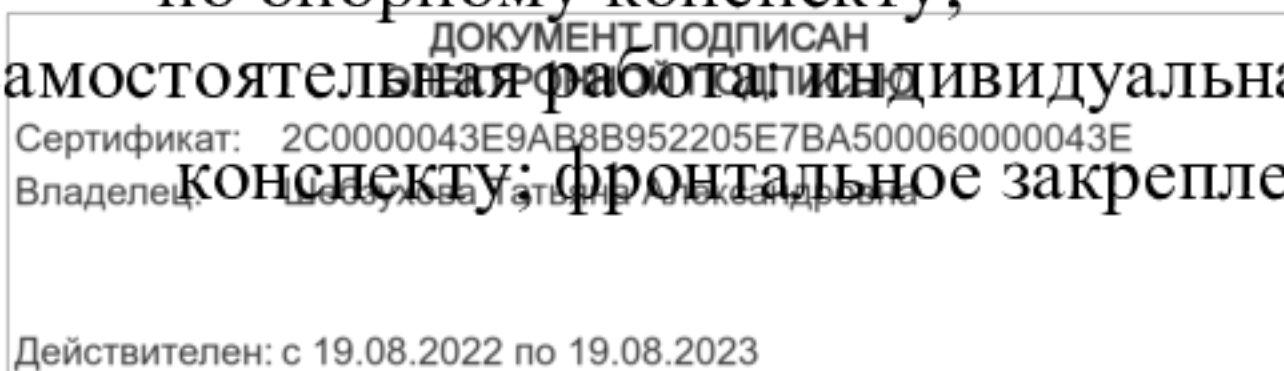
Основой технологии интенсификации обучения на платформе цифровых образовательных технологий являются учебно-иллюстрационные материалы (опорный конспект) по дисциплине «Управление данными».

Работа с учебно-иллюстрационными материалами имеет следующие этапы.

Изучение теоретических основ учебного материала в аудитории:

изложение преподавателем изучаемого материала студентам с объяснением по опорному конспекту;

Самостоятельная работа индивидуальная работа студентов по опорному конспекту; фронтальное закрепление по блокам опорного конспекта.



Первое повторение - воспроизведение содержания заданной темы опорного конспекта по памяти.

Устное проговаривание материала опорного конспекта - необходимый этап внешнеречевой деятельности при усвоении учебного материала.

Второе повторение - взаимопрос и взаимопомощь студентов друг другу.

Применение учебно-иллюстрационных материалов позволяет обобщить сложный по содержанию материал, активизировать мыслительную деятельность студентов.

Необходимо помнить, что главное для студента в самостоятельной работе с рекомендуемой литературой и источниками - это формирование своего индивидуального стиля, который может стать основой в будущей профессиональной деятельности.

7.2. Перечень рекомендуемой литературы

7.1.1. Основная литература:

Основная литература

1. Швецов, В.И. Базы данных Электронный ресурс : учебное пособие / В.И.

Швецов. - Базы данных, 2019-12-01. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 218 с. - Книга находится в базовой версии ЭБС IPRbooks.

2. Крис, Файли . SQL Электронный ресурс / Файли Крис ; пер. А. В. Хаванов. - SQL, 2019-04-19. - Саратов : Профобразование, 2017. - 452 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-44880103-7

3. Дроздова, В. И. (Северо-Кавказский федеральный университет). Управление данными : учеб. пособие : Направление подготовки 320400.62- Информационные системы и технологии. Бакалавриат / В. И. Дроздова ; Сев.Кав. федер. ун-т- Ставрополь : СКФУ, 2013. - 170 с.

7.1.2. Дополнительная литература:

0 Агальцов, В. П. Базы данных : учебник / В. П. Агальцов, Кн.2, Распределенные и удаленные базы данных. - М. : Форум : ИНФРА-М, 2014. -

с. : ил. - (Высшее образование). - Гриф: Доп. УМО. - Библиогр.: с.

-

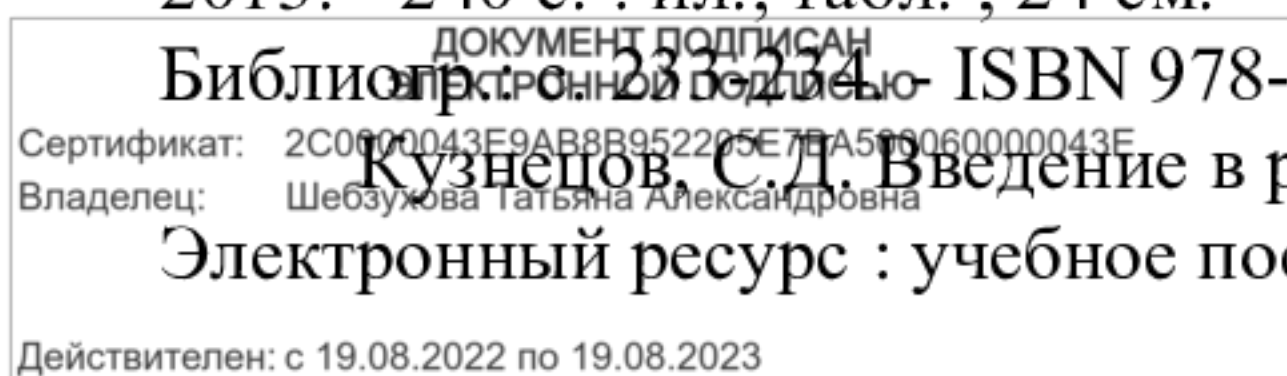
Карпова, И. П. Базы данных : курс лекций и материалы для практических занятий : учебное пособие / И. П. Карпова. - СПб. [и др.] : Питер,

2013. - 240 с. : ил., табл. ; 24 см. - (Учебное пособие). - Гриф: Рек. -

Библиогр. с. 233-234. - ISBN 978-5-496-00546-3

Кузнецов, С.Д. Введение в реляционные базы данных

Электронный ресурс : учебное пособие / С.Д. Кузнецов. - Введение в



реляционные базы данных, 2021-01-23. - Москва : Интернет-Университет
Информационных

Технологий (ИНТУИТ), 2016. - 247 с. - Книга находится в базовой версии
ЭБС IPRbooks. - ISBN 5-9556-00028-0

7.1.3. Интернет-ресурсы:

<http://www.citforum.ru/database/>

<http://www.acm.org/sigmod/publications.html>

<http://www.acm.org/sigmod/databaseSoftware/index.html>

Материально-техническое обеспечение дисциплины

индивидуальное взаимодействие со студентами по электронной почте для
предварительного ознакомления с их разработками при подготовке к
аудиторным занятиям;

использование на лекциях и лабораторных занятиях мультимедийного
оборудования для демонстрации листингов программного кода, скриншотов
выполненных запросов, работы программ и пр.;

включение в лабораторные работы индивидуального поиска, систематизации

0 анализа информации через Интернет;

авторские презентации к лекциям;

лекционная аудитория должна быть оборудована мультимедиа
проектором.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ
КОНТРОЛЬНОЙ РАБОТЫ
ПО ДИСЦИПЛИНЕ
УПРАВЛЕНИЕ ДАННЫМИ**

Направление подготовки

09.03.02

**Информационные системы и
технологии**

Квалификация выпускника

Бакалавр

Пятигорск, 2023

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Содержание

Введение.....	3
1. Цель, задачи и реализуемые компетенции.....	3
2. Формулировка задания и его объем.....	3
3. Общие требования к написанию и оформлению работы.....	3
4. Варианты заданий для студентов заочной формы обучения.....	3
5. План-график выполнения задания.....	16
6. Критерии оценивания работы.....	17
7. Порядок защиты работы.....	17
8. Учебно-методическое и информационное обеспечение дисциплины.....	17

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Введение

Методические указания содержат перечень вариантов заданий для контрольных работ, требования к оформлению контрольных работ и пример выполнения задания. Теоретической основой подготовки специалиста являются знания в области информатики, вычислительной систем.

Цель, задачи и реализуемые компетенции

Методические указания составлены с учетом требований стандарта высшего образования по дисциплине: «Управление данными».

Целью изучения дисциплины является приобретение знаний об архитектурах баз данных, основных моделях баз данных, языках описания и манипулирования данными; современных методах проектирования реляционных баз данных, одной из конкретных СУБД MS SQL Server.

Задачи изучения дисциплины: формирование представлений об архитектуре баз данных, моделях баз данных; основных конструкциях языков описания и манипулирования данными, этапах проектирования баз данных, методах проектирования реляционных баз данных, подходах к составлению приложений для базы данных, способах обеспечения защиты и целостности баз данных.

Формулировка задания и его объем

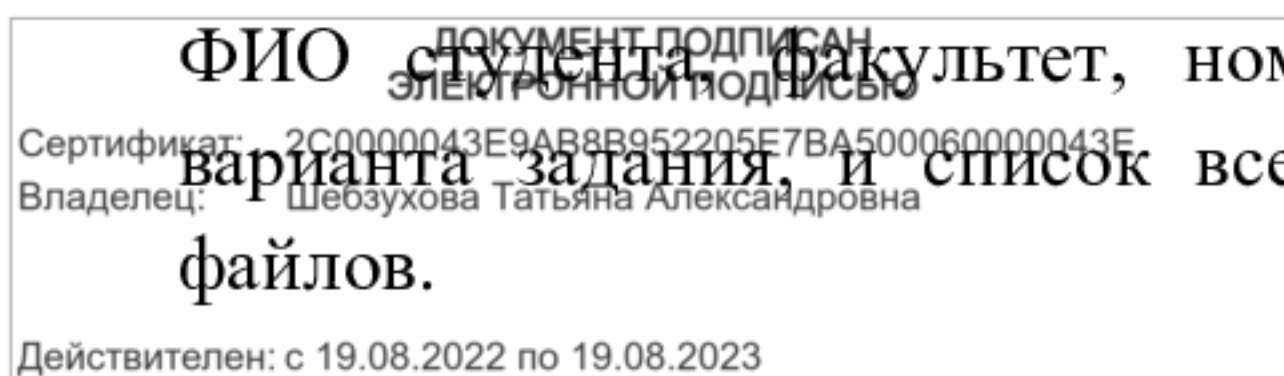
Контрольная работа включает в себе пять заданий по темам, которые нужно изучить самостоятельно и по ним подготовить контрольную работу. Результаты выполнения контрольной работы предоставляются в электронном виде. Объем контрольной работы составляет 15-20 печатных листов формата А4.

Вариант задания выбирается по двум последним цифрам зачетной книжки из каждого предложенного задания.

Общие требования к написанию и оформлению работы

Контрольная работа выполняется и сдается в электронном виде на CD/CDRW носителе. На конверте необходимо указать название дисциплины,

ФИО студента, факультет, номер группы, шифр зачетной книжки, № варианта задания, и список всех созданных в ходе выполнения задания файлов.



Приведенный в конце методических указаний список литературы может использоваться студентами при выполнении контрольной работы.

Варианты заданий для студентов заочной формы обучения

Задание 1. Разработка структуры БД, ввод и корректировка БД.

Порядок выполнения работы:

создать таблицу и ввести в нее данные,
выполнить просмотр и корректировку созданной структуры и введенных данных,
осуществлять редактирование, вставку и удаление записей таблицы посредством меню,

Каждый студент выполняет вариант согласно порядковому номеру в журнале.

Количество записей, введенных в базу данных, должно быть не меньше 15.

Варианты заданий

Вариант 1

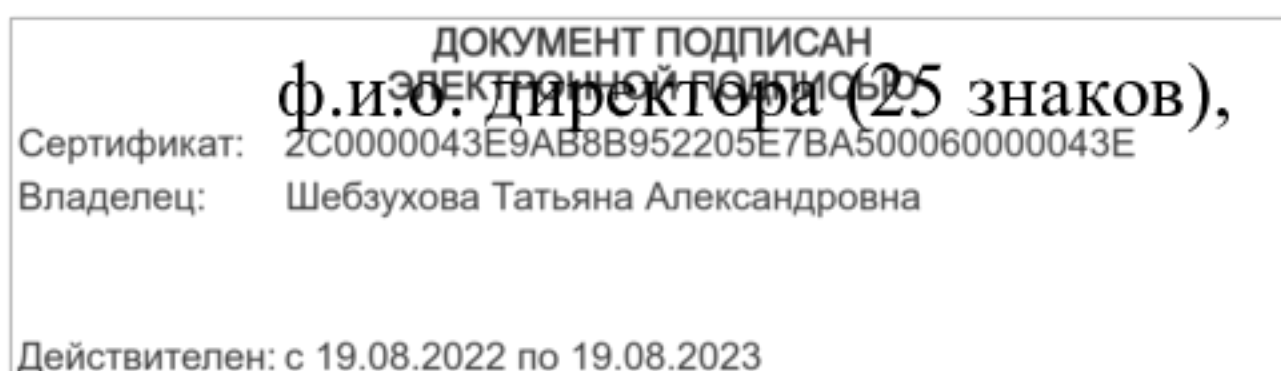
проданных товарах имеется следующая информация: -
артикул товара (5 знаков), - наименование товара (20 знаков), - дата начала продажи товара,
- количество проданного товара (4 знака),
- цена за единицу товара (в тенге и тиынах 6 знаков)

Вариант 2

0 расходовании электроэнергии на заводах имеется следующая информация:

номер завода (4 знака),

ф.и.о. директора (25 знаков),



0 ф.и.о. главного энергетика (25 знаков),

1 дата отчета,

2 количество израсходованной электроэнергии (в тыс. кВт*ч, 5 знаков)

Вариант 3 Даны сведения о заболеваемости

студентов:

3 наименование группы (8 знаков),

4 факультет (6 знаков),

5 дата медосмотра,

6 фамилия студента (25 знаков),

код диагноза (5 знаков)

Вариант 4

Даны сведения о рабочих завода:

0 код предприятия (4 знака),

1 номер цеха (2 знака),

2 табельный номер рабочего (5 знаков),

3 дата поступления на работу,

пол (1 знак)

Вариант 5

Имеется следующая информация о результатах отладки программ для курсовых проектов:

0 шифр группы (8 знаков),

1 наименование дисциплины (20 знаков),

2 дата отчета,

3 количество студентов в группе (2 знака),

число отлаженных программ (2 знака)

Вариант 6

Имеется следующая информация о наличии технических средств на добычных участках:

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат:	2C00000043E9AB8B952205E7BA5000600000043E
Владелец:	Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023	

0 дата отчета,
1 номер участка (2 знака),

количество продавцов в отделе (2 знака),
средняя заработная плата по отделу (в долларах 3 знака).

Вариант 11 Имеется следующая информация о студентах,

занимающихся спортом:

фамилия студента (25 знаков),
шифр группы (8 знаков),
код вида спорта (2 знака),
наименование вида спорта (20 знаков),
разряд (1 знак).

Вариант 12

Имеется следующая информация о прохождении студентами практики на предприятиях:

страна (20 знаков),
название города (20 знаков),
наименование предприятия (20 знаков),
код предприятия (4 знака),
шифр группы (8 знаков),
ф.и.о. студента (20 знаков).

Вариант 13

Имеется следующая информация о пропусках занятий студентами:

шифр группы (8 знаков),
фамилия студента (20 знаков),
пропущено часов (3 знака),
оправдано часов (3 знака).

Вариант 14 Имеется следующая информация о сдаче

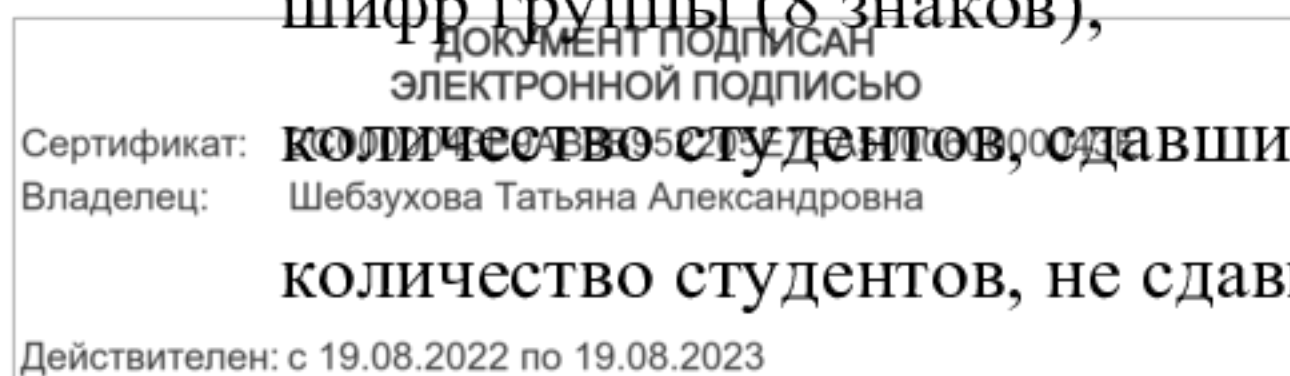
студентами сессии:

сессия (зимняя, весенняя - 10 знаков),

шифр группы (8 знаков),

количество студентов, сдавших сессию досрочно (3 знака),

количество студентов, не сдавших сессию в срок (3 знака),



количество студентов, сдавших сессию вовремя (3 знака).

Вариант 15 Имеется следующая информация о

породах собак:

фамилия владельца (25 знаков),

код породы (3 знака),

наименование породы (25 знаков),

количество медалей (2 знака),

наименование клуба (25 знаков).

Вариант 16

Имеется следующая информация о ПЭВМ:

код предприятия (5 знаков),

наименование предприятия (20 знаков),

тип компьютера (20 знаков),

количество компьютеров (2 знака),

страна-изготовитель (25 знаков).

Вариант 17 Имеется следующая информация о предметах, изучаемых в семестре:

шифр группы (8 знаков),

количество предметов (2 знака),

количество экзаменов (2 знака),

количество зачетов (2 знака),

количество курсовых проектов (1 знак).

Вариант 18

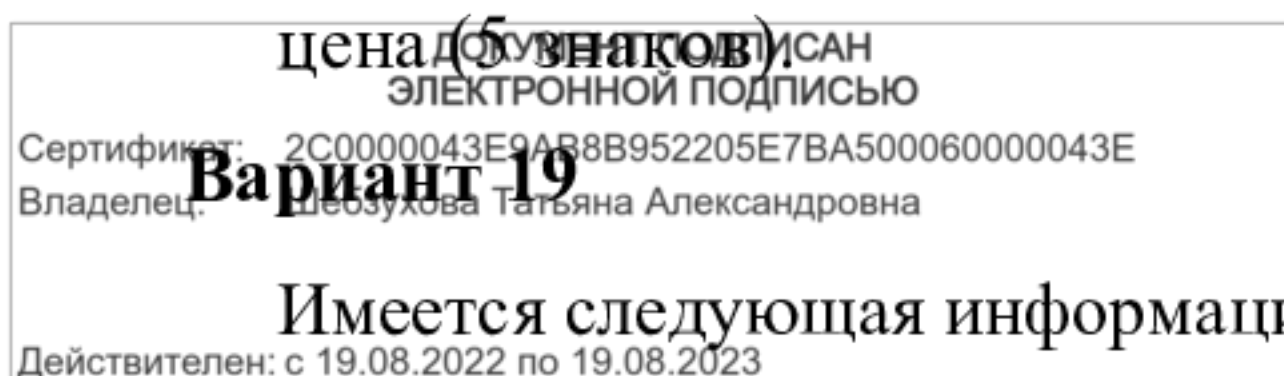
Имеется следующая информация о сделанных покупках:

наименование покупки (25 знаков),

наименование магазина (25 знаков),

количество (5 знаков),

цена (5 знаков).



фамилия больного (25 знаков),
фамилия врача (25 знаков),
наименование операции (25 знаков),
код операции (2 знака),
номер палаты (2 знака).

Вариант 20

Имеется следующая информация об использовании ПЭВМ в ауд.430:

номер машины (2 знака),
тип компьютера (20 знаков),
дата (8 знаков),
время загрузки (2 знака).

Вариант 21

Имеется следующая информация о распределении студентов:

фамилия студента (25 знаков),
шифр группы (8 знаков),
наименование предприятия (25 знаков),
код предприятия (4 знака),
средний балл в приложении к диплому (2 знака).

Вариант 22

Имеется следующая информация о кооперативах в нашем городе:

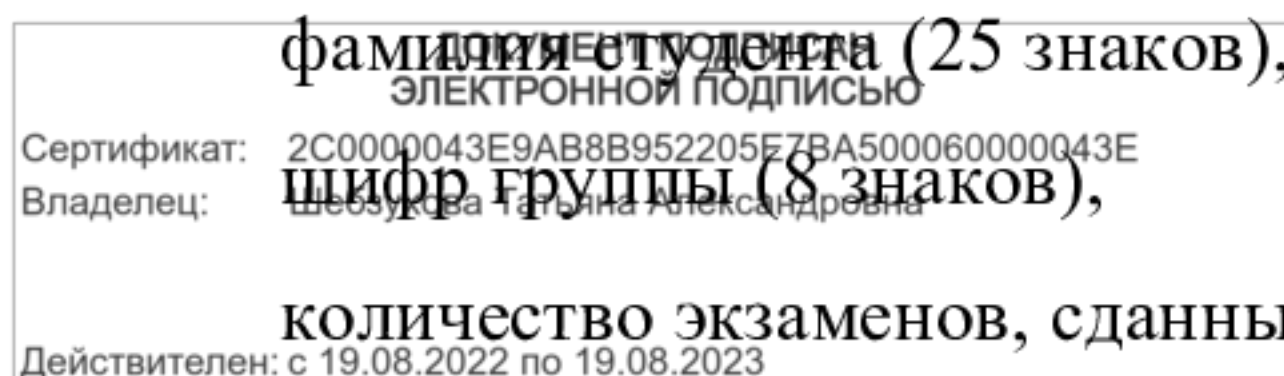
0 фамилия председателя (25 знаков),
1 шифр кооператива (3 знака),
2 наименование кооператива (25 знаков),
3 тип (4 знака - строительный, швейный и т.д.),
годовая прибыль (7 знаков).

Вариант 23

Имеется следующая информация о составе групп:

фамилия студента (25 знаков),
шифр группы (8 знаков),

количество экзаменов, сданных на 5 (1 знак),



количество экзаменов, сданных на 4 (1 знак),

количество экзаменов, сданных на 3 (1 знак).

Вариант 24 Имеется следующая информация о погоде

на квартал:

общее количество дней (3 знака),

количество солнечных дней (3 знака),

количество пасмурных дней (3 знака),

фамилия метеоролога (25 знаков),

средняя температура (2 знака).

Вариант 25

Имеется следующая информация о распределении мест на теннисных соревнованиях:

0 фамилия теннисиста (25 знаков):

1 шифр группы (2 знака),

2 количество выигрышей (3 знака),

3 количество проигрышей (3 знака),

место, занимаемое в общем списке (3 знака).

Контрольные вопросы:

Основные функции СУБД.

Что такое БД?

Назначение таблиц?

Управление таблицами

Отношения между таблицами

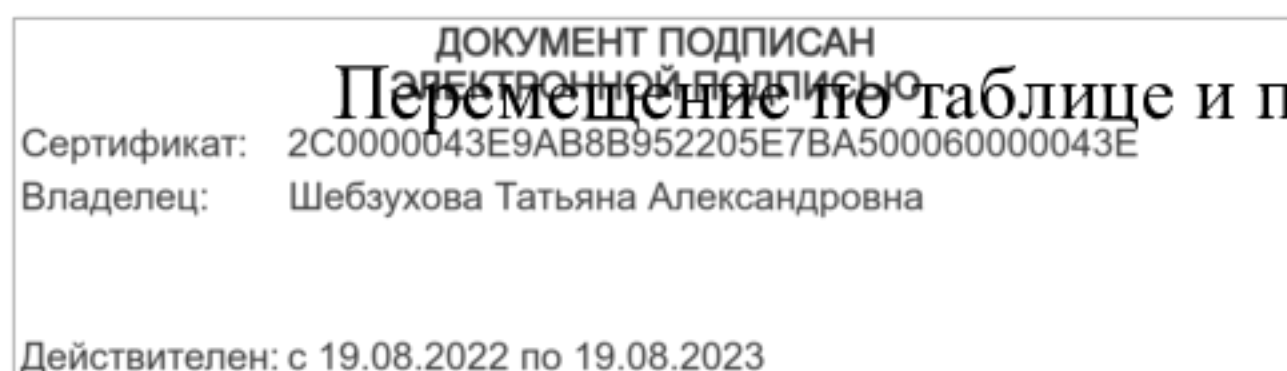
Команды добавления, удаления, поиска, перемещения по записям.

Отличие простого индекса от составного. Различные типы индексов.

Контрольные задания для СРС.

Создание простых и составных индексов.

Перемещение по таблице и поиск данных по определенному критерию.



Задание 2. Организация циклов. Индексные файлы.

Порядок выполнения работы:

Создать две таблицы и ввести в них данные, используя маски ввода, свойство значения по умолчанию согласно варианту (в качестве ключа назначьте данные, выделенные символом подчеркивания).

Определите условия целостности и достоверности вводимых данных.

Варианты заданий

Вариант 1

Код организации, наименование организации

Код организации, дата отгрузки, кол-во товара, артикул товара, номер
накладной

Вариант 2

Код цеха, наименование цеха, ФИО начальника _____

Таб. номер, код цеха, кол-во отработанных часов

Вариант 3

Код сбербанка, наименование сбербанка

Таб. номер сотрудника, Код сбербанка, Лицевой счет, % перечисления

Вариант 4

Код цеха, наименование цеха, ФИО начальника

Таб. Номер, код цеха, кол-во отработанных дней

Вариант 5

Код организации, наименование организации, ФИО экспедитора, номер машины, марка машины

накладной, дата, код организации, артикул товара, кол-во товара,
признак (отгрузка=1/прием=0)

Вариант 6

Дата продажи, дата отправления, ФИО пассажира, кол-во билетов, № документа

ЭЛЕКТРОННОЙ ПОДПИСЬЮ
рейса, номер билета

Сертификат: 2000000459 рейса номер билета 0000000000000000000000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Вариант 7

Код продукта, наименование продукта_

Код блюда, код продукта, кол-во продукта

заказа, код блюда, кол-во порций, ФИО

Вариант 8

Код работника, ФИО получателя платежа, адрес получателя, процент отчислений

Код цеха, наименование

Вариант 9

Индекс издания, Наименование издания, цена

Код подписчика, индекс издания, дата подписки, период подписки, кол-во экземпляров

Вариант 10

накладной, код товара, кол-во, дата, код склада_

Код склада, наименование, ФИО кладовщика

Вариант 11

Дата поступления, код товара, № склада, кол-во, инвент.номер

склада, наименование, ФИО кладовщика

Вариант 12

Код книги, наименование, ФИО автора

Код читателя, код книги, дата выдачи, дата возврата

Вариант 13

Код читателя, ФИО, адрес, № телефона, дата рождения

Код книги, код читателя, дата выдачи, дата возврата_

Код автора, ФИО автора

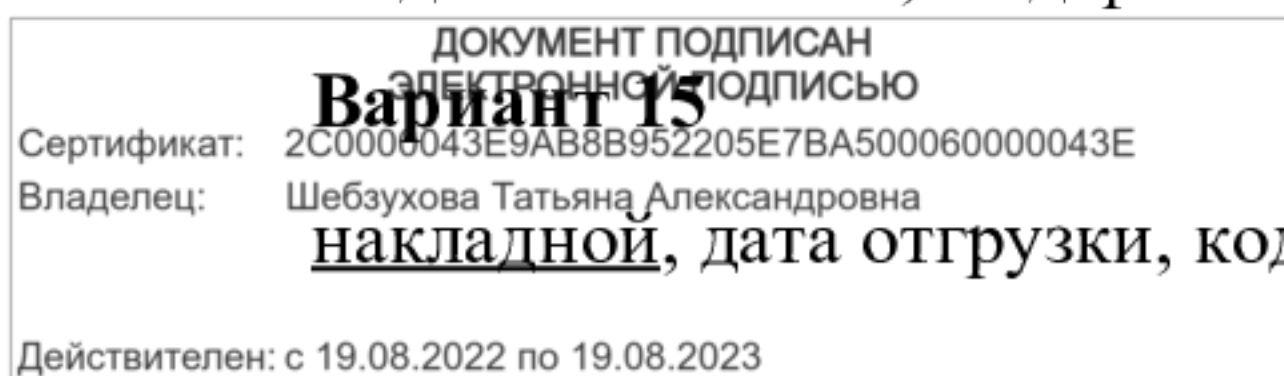
Вариант 14

Код пользователя, ФИО, адрес, пол

Код пользователя, Код фильма, дата выдачи, дата возврата, сумма залога

Вариант 15

накладной, дата отгрузки, код водителя, кол-во товара, код товара



Код товара, цена за ед. измерения, ед. измерения

Вариант 16

Код пользователя, Код диска, дата выдачи, дата возврата

Код пользователя, ФИО, адрес, телефон, пол, сумма залога

Вариант 17

Код врача, ФИО врача, специальность

Дата назначения, код врача, код лекарства, ФИО больного

Вариант 18

накладной, код организации, код товара, кол-во

Код товара, наименование, ед. измерения, цена

Код банка, наименование банка

Вариант 19

Код врача, ФИО врача, специальность врача

Код пациента, код врача, дата приема, время приема

Вариант 20

Код жилья, ФИО, адрес

Код жилья, код издания, кол-во экземпляров

Код автора, наименование автора

Вариант 21

Код закройщика, ФИО закройщика

Код изделия, наименование изделия, цена, тип изделия (верхняя одежда, легкое платье...)

Вариант 22

Код страхуемого лица, ФИО, адрес, код страховки, дата страховки, код страховщика

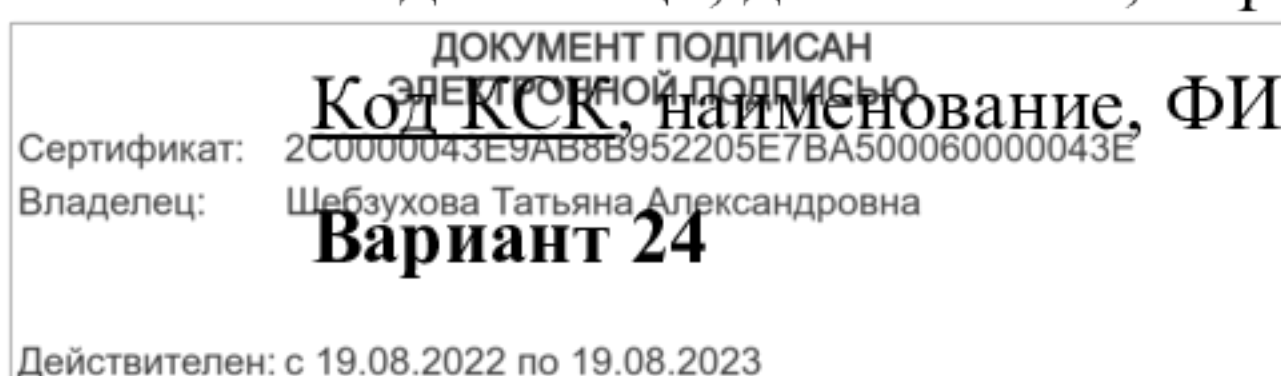
Код страховщика, ФИО

Вариант 23

Код жилья, дата взноса, период взноса, сумма, код КСК

Код КСК, наименование, ФИО бухгалтера

Вариант 24



Код налоговой инспекции, наименование

Код налоговой инспекции, код налогоплательщика, дата

Вариант 25

Номер автомобиля, код марки автомобиля, номер двигателя, номер шасси, код цвета

Код цвета, наименование цвета

Код марки автомобиля, наименование

Контрольные вопросы:

Понятие индекса. Виды индексов.

Способы сортировки информации.

Основные типы связей между таблицами.

Что такое ключевые поля?

Как установить (разорвать) связи между таблицами?

Способы определения условий целостности.

Механизмы определения условий достоверности данных.

Для чего используется представление данных?

Способы редактирования данных при помощи представлений.

Контрольные задания для СРС.

На основании таблицы создать представление для редактирования одного из полей таблицы в режиме просмотра.

Задание 3. Объекты и управление объектами СУБД. Работа с таблицами БД.

Порядок выполнения работы:

Создать форму, содержащую элементы управления и реализовать с помощью них просмотр и редактирование данных из задания №1;

Осуществить экспорт одной из таблиц вашей БД в файл формата Microsoft Excel;

Создать какую-либо таблицу в Microsoft Excel и проимпортировать эти данные в вашу базу данных;

Внедрить объект OLE (по вашему усмотрению) в таблицу БД;

Варианты заданий

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат:	2C0000043E9AB8B952205E7BA500060000043E
Владелец:	Шарова Александра Александровна
Действителен: с 19.08.2022 по 19.08.2023	

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество товара и его стоимость.

Вариант 2

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество заводов и расход энергии.

Вариант 3

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество больных студентов и общее число дней нетрудоспособности.

Вариант 4

Вывести значения всех полей базы данных на экран.
Подсчитать итог: количество женщин и мужчин.

Вариант 5

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество студентов и общее число отлаженных программ.

Вариант 6

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество каждого вида оборудования.

Вариант 7

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество автомашин и расход топлива.

Вариант 8

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество маршрутов и подвижного состава.

Вариант 9

Вывести значения всех полей базы данных на экран.
Подсчитать итог: общее количество рейсов и самолетов.

Вариант 10

Вывести значения всех полей базы данных на экран.
Подсчитать итог: количество продавцов и общую сумму месячной заработной платы.

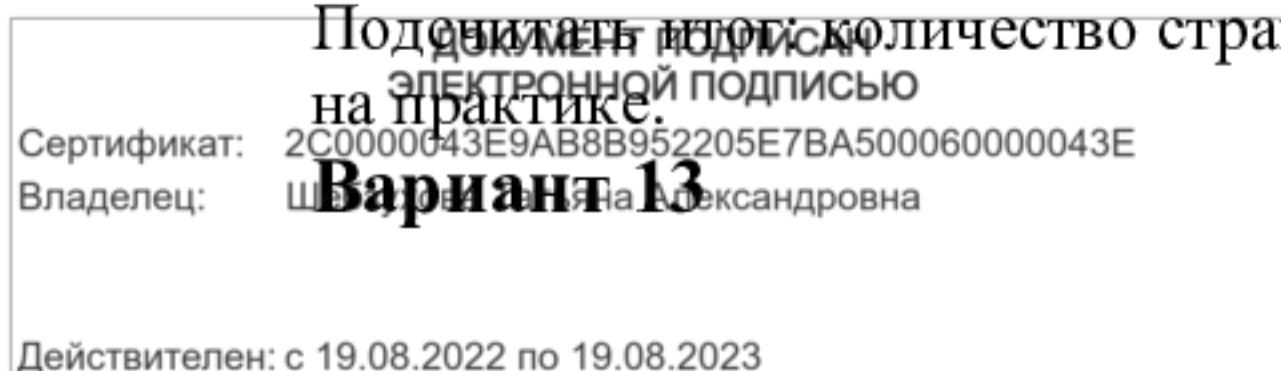
Вариант 11

Вывести значения всех полей базы данных на экран.
Подсчитать итог: количество студентов, занимающихся спортом и количество видов спорта.

Вариант 12

Вывести значения всех полей базы данных на экран.
Подсчитать итог: количество стран или количество городов и количество студентов на практике.

Вариант 13



Вывести значения всех полей базы данных на экран.
Подсчитать итог: количество студентов и количество пропущенных часов.

Вариант 14

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество студентов и количество групп.

Вариант 15

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество пород и общее количество собак.

Вариант 16

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество предприятий и компьютеров.

Вариант 17

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество групп и среднее количество экзаменов.

Вариант 18

Вывести значения всех полей базы данных на экран.

Подсчитать итог: общее количество покупок и их стоимость.

Вариант 19

Вывести значения всех полей базы данных на экран.

Подсчитать итог: общее количество больных и врачей.

Вариант 20

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество машин и общее время загрузки.

Вариант 21

Вывести значения всех полей базы данных на экран.

Подсчитать итог: общее количество студентов и предприятий.

Вариант 22

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество кооперативов и общая годовая прибыль.

Вариант 23

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество групп и студентов.

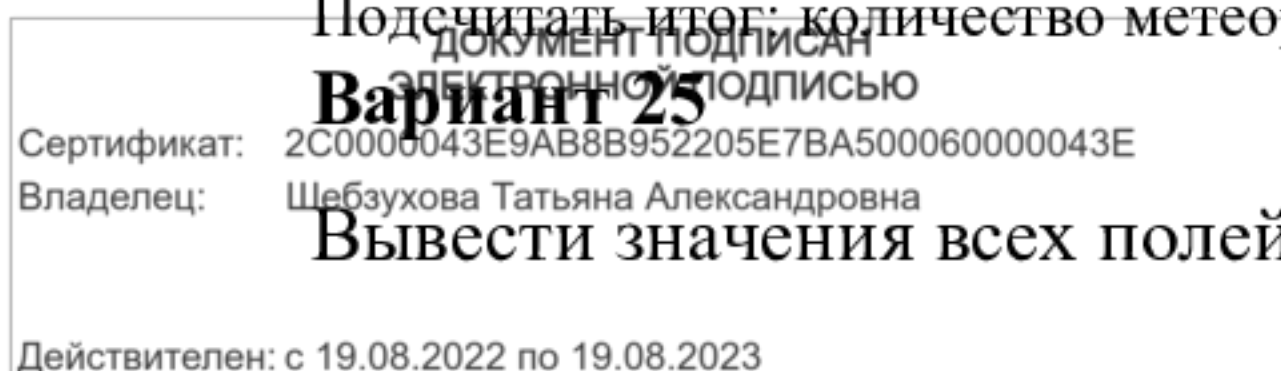
Вариант 24

Вывести значения всех полей базы данных на экран.

Подсчитать итог: количество метеорологов и среднее количество пасмурных дней.

Вариант 25

Вывести значения всех полей базы данных на экран.



Подсчитать итог: количество спортсменов и общее количество игр.

Контрольные вопросы:

Что такое объекты в MS SQL Server?

Чем отличаются друг от друга контейнеры и средства управления?

Какими путями можно создавать объекты?

Как осуществляется доступ к свойствам объекта?

Каким образом можно связать объект с данными?

Что могут содержать в себе контейнеры?

Какие объекты менее доступны для модификации и менее гибки?

Объясните понятия метода и события объекта?

Основные способы работы с файлами других форматов данных?

С какими форматами и базами данных может взаимодействовать?

Как связать OLE-объект с объектом на сервере, какие способы существуют?

Как осуществляется связывание объектов?

Как осуществляется внедрение объектов?

Как осуществляется экспорт объектов?

Как осуществляется импорт объектов?

Назначение макроса. Способы его создания

Контрольные задания для СРС.

Осуществить экспорт таблицы вашей БД в файл формата Microsoft Word;
Создать макрос.

Задание 4. Запросы в SQL. Работа с запросами

Порядок выполнения работы:

Используя базу данных, созданную в задании №1:

Создать запрос на выборку, согласно индивидуальному варианту

Создайте запрос по своему усмотрению с вычисляемым полем на основании данных, взятых из созданных вами ранее таблиц.

Варианты заданий

Вариант 1

Результат запроса должен содержать всю имеющуюся информацию о товарах, проданных после определённой в запросе даты.

ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Вариант 2

Сертификат: 2C0006043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Результат запроса должен содержать информацию о номерах заводов, расход электроэнергии на которых превысил определённый лимит, и о количестве израсходованной электроэнергии.

Вариант 3

Результат запроса должен содержать все имеющиеся сведения о студентах, прошедших медосмотр в определённый промежуток времени.

Вариант 4

Результат запроса должен содержать все имеющиеся сведения о рабочих – мужчинах, поступивших на работу позднее определённого срока.

Вариант 5

Результат запроса должен содержать информацию о результатах отладки программ по какой-либо одной дисциплине.

Вариант 6

Результат запроса должен содержать дату отчета, номер участка и количество комбайнов для всех участков, данные отсортировать по участкам.

Вариант 7

Результат запроса должен содержать номер колонны, марку автомашины и расход топлива только тех колонн, расход топлива в которых превысил какое-то определенное значение.

Вариант 8

Результат запроса должен содержать все имеющиеся данные о маршрутах, эксплуатация которых началась раньше определённого срока.

Вариант 9

Результат запроса должен содержать информацию об авиарейсах, кроме стоимости билета, которые были введены в эксплуатацию раньше определенного срока.

Вариант 10

Результат запроса должен содержать всю имеющуюся информацию о сотрудниках магазина, работающих в отделах, где количество продавцов больше 1.

Вариант 11

Результат запроса должен содержать всю имеющуюся информацию о студентах, занимающихся спортом и имеющих первый спортивный разряд.

Вариант 12

Сертификат:	2C0000043E9AB8B952205E7BA5000600000043E
Владелец:	Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023	

Результат запроса должен содержать всю имеющуюся информацию о прохождении студентами практики за пределами Казахстана.

Вариант 13

Результат запроса должен содержать всю имеющуюся информацию о студентах, разница между пропущенными и оправданными часами которых больше определённого числа.

Вариант 14

Результат запроса должен содержать всю имеющуюся информацию о сдаче студентами весенней сессии.

Вариант 15

Результат запроса должен содержать всю имеющуюся информацию о собаках определённой породы.

Вариант 16

Результат запроса должен содержать всю имеющуюся информацию о ПЭВМ на предприятии, имеющем определённый код.

Вариант 17

Результат запроса должен содержать всю имеющуюся информацию о предметах для определённой группы.

Вариант 18

Результат запроса должен содержать всю имеющуюся информацию о покупках, сделанных в каком-либо одном магазине.

Вариант 19

Результат запроса должен содержать всю имеющуюся информацию об операциях, каким-либо одним врачом.

Вариант 20

Результат запроса должен содержать всю имеющуюся информацию об использовании ПЭВМ в ауд.430, время загрузки которых превышает 90 минут.

Вариант 21

Результат запроса должен содержать всю имеющуюся информацию о распределении студентов какой-либо одной группы.

Вариант 22

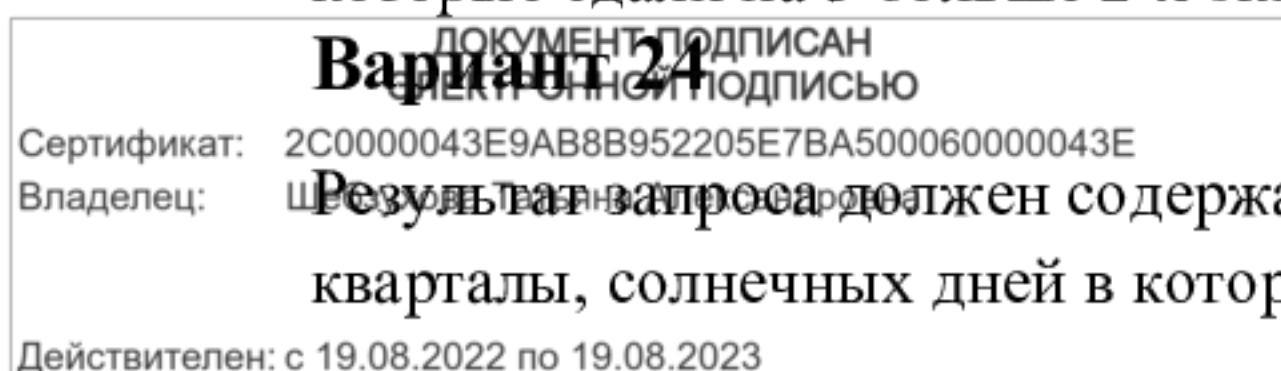
Результат запроса должен содержать всю имеющуюся информацию о кооперативах, прибыль которых превысила определённую сумму.

Вариант 23

Результат запроса должен содержать всю имеющуюся информацию о студентах, которые сдали на 3 больше 2-х экзаменов.

Вариант 24

Результат запроса должен содержать всю имеющуюся информацию о погоде за кварталы, солнечных дней в которых было больше определённого количества.



Вариант 25

Результат запроса должен содержать всю имеющуюся информацию о теннисистах, занимающих три первых места в турнирной таблице.

Контрольные вопросы:

Для чего предназначен SQL?

В каком случае игнорируется опция TO команды SELECT?

Объясните разницу между клиентом и сервером.

Чем отличаются локальные и удалённые виды?

Назначение запросов?

Основные виды запросов?

Как можно создать запрос?

Как определяются заголовки столбцов, используемых в работе с запросом?

Может ли вид включать в себя другие виды?

10. Для чего используются вычисляемые поля?

11. Как произвести итоговые вычисления по группам записей?

Контрольные задания для СРС.

Создать запрос по своему усмотрению на основании данных, взятых из созданных ранее таблиц с расчетом итоговых значений.

Задание 5. Работа с формами, отчётами, меню и панелями инструментов.

Порядок выполнения работы:

Используя базу данных, созданную в лабораторной работе №1, необходимо:

Отредактировать созданную форму, (должны быть реализованы следующие функции: ввод записей в уже существующую таблицу, просмотр данных, корректировка отдельных полей, удаление).

Создать меню, дублирующее панель инструментов на форме.

Создать отчет согласно индивидуальному варианту в табличном виде на основании таблицы, созданной в лабораторной работе 1.

Отчеты должны содержать заголовок, дату и время создания, сопроводительный текст (автор отчета), информацию о количестве страниц и вычисление итоговых результатов, где это возможно.

Варианты заданий

ДОКУМЕНТ ПОДПИСАН
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Вариант 1

Отчёт должен содержать все данные о товарах и итоговую сумму продаж.

Вариант 2

Отчёт должен содержать все данные о расходовании электроэнергии на заводах, и итоговую сумму расходуемой электроэнергии.

Вариант 3

Отчёт должен содержать все данные о заболеваемости студентов и количество студентов, прошедших медосмотр по каждому факультету.

Вариант 4

Отчёт должен содержать все данные о рабочих и количество рабочих в каждом цехе.

Вариант 5

Отчёт должен содержать все данные о результатах отладки программ и число отлаженных программ в каждой группе.

Вариант 6

Отчёт должен содержать все данные о наличии технических средств на добычных участках и общее количество комбайнов, бункеров-перегрузателей и самоходных вагонов.

Вариант 7

Отчёт должен содержать все данные о расходовании топлива в автоколоннах и итоговую сумму расхода топлива.

Вариант 8

Отчёт должен содержать все данные о городских маршрутах и общее количество троллейбусов, трамваев и автобусов.

Вариант 9

Отчёт должен содержать все данные об авиарейсах и количество самолётов каждой марки в отдельности, совершающих рейсы.

Вариант 10

Отчёт должен содержать все данные о сотрудниках магазина ЦУМ и общее количество продавцов.

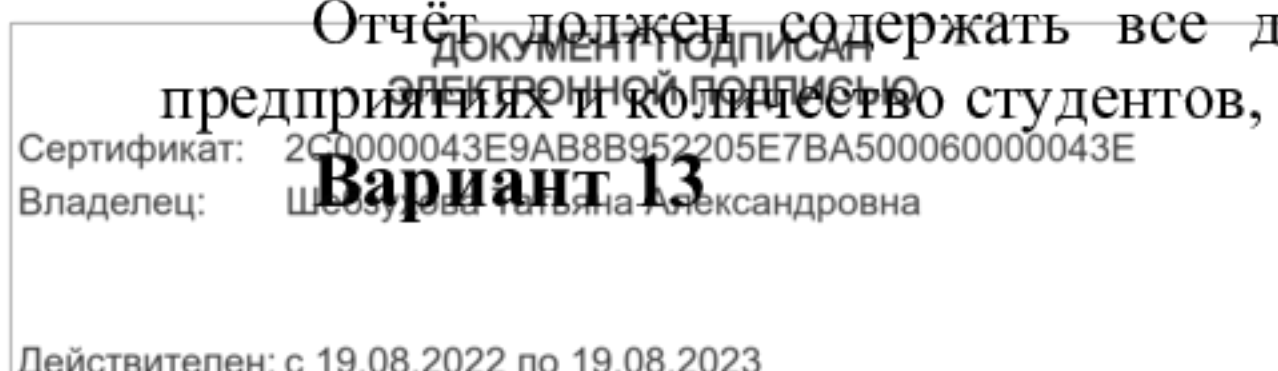
Вариант 11

Отчёт должен содержать все данные о студентах, занимающихся спортом и их общее количество.

Вариант 12

Отчёт должен содержать все данные о прохождении студентами практики на предприятиях и количество студентов, проходящих практику на каждом предприятии.

Вариант 13



Отчёт должен содержать все данные о пропусках занятий студентами и количество пропусков по каждой группе.

Вариант 14

Отчёт должен содержать все данные о сдаче студентами сессии и количество студентов, сдавших каждую сессию досрочно.

Вариант 15

Отчёт должен содержать все данные о породах собак и количество собак каждой породы.

Вариант 16

Отчёт должен содержать все данные о ПЭВМ и общее количество компьютеров каждого типа.

Вариант 17

Отчёт должен содержать все данные о предметах, изучаемых в семестре

Вариант 18

Отчёт должен содержать все данные о сделанных покупках и общее количество покупок каждого наименования.

Вариант 19

Отчёт должен содержать все данные об операциях, сделанных в больнице и количество операций каждого наименования.

Вариант 20

Отчёт должен содержать все данные об использовании ПЭВМ в ауд.430 и количество компьютеров каждого типа.

Вариант 21

Отчёт должен содержать все данные о распределении студентов и количество студентов по каждому предприятию.

Вариант 22

Отчёт должен содержать все данные о кооперативах в нашем городе и количество кооперативов каждого типа.

Вариант 23

Отчёт должен содержать все данные о составе групп и количество студентов в каждой группе.

Вариант 24

Отчёт должен содержать все данные о погоде на квартал, общее количество солнечных дней во всех кварталах, общее количество пасмурных дней во всех кварталах.

Вариант 25

Отчёт должен содержать все данные о распределении мест на теннисных

соревнованиях

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Контрольные вопросы:

Действителен: с 19.08.2022 по 19.08.2023

Назначение форм.
 Способы создания форм.
 В чем заключается различие между формами и представлением данных в режиме таблицы?
 Что такое среда данных?
 Что такое элементы управления?
 Основные типы элементов управления?
 Как определяются свойства элементов управления?
 Как добавить в форму элемент управления?
 Назначение и типы отчетов.
 Способы создания отчетов.
 Типы полос конструктора отчетов и их назначение.
 Вычисляемые и итоговые поля в отчетах
 Группировка данных в отчетах
 Что включает в себя система меню?
 Как определяются процедуры для пунктов меню?
 Отличие «горячих клавиш» меню от клавиш ускоренного действия

Контрольные задания для СРС.

Создать отчет в свободной форме на основании одного из запросов, созданных в задании 3.

0 План-график выполнения задания

Дата получения задания	Дата предоставления выполненного задания
Установочная сессия.	Экзаменационная сессия за две недели до начала.

Критерии оценивания работы

Оценка «отлично» выставляется студенту, если он продемонстрировал глубокие, исчерпывающие знания и творческие способности в понимании, изложении и использовании учебно-программного материала; логически последовательные, содержательные, полные, правильные и конкретные ответы на все поставленные вопросы и дополнительные вопросы преподавателя; свободное владение основной и дополнительной литературой, рекомендованной учебной программой.

Оценка «хорошо» выставляется студенту, если он продемонстрировал твердые и достаточно полные знания всего программного материала, правильное понимание сущности и взаимосвязи рассматриваемых процессов явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении замечаний по отдельным вопросам; достаточное владение литературой, рекомендованной учебной программой.

ДОКУМЕНТ ПОДПИСАН
 Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
 Владелец: 2C0000043E9AB8B952205E7BA500060000043E
 Действителен: с 19.08.2022 по 19.08.2023

Оценка «удовлетворительно» выставляется студенту, если он продемонстрировал твердые знания и понимание основного программного материала; правильные, без грубых ошибок ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточное владение литературой, рекомендованной учебной программой.

Оценка «неудовлетворительно» выставляется студенту, если он продемонстрировал неправильные ответы на основные вопросы, допущены грубые ошибки в ответах, непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы.

Порядок защиты работы

Защита контрольной работы проводится в виде научного диспута с презентацией выполненных заданий, в соответствии с графиком защиты. После доклада студенту задаются вопросы как преподавателем, так и студентами группы.

В процессе защиты своей работы студент делает доклад продолжительностью 7-10 минут. Доклад должен быть предварительно подготовлен студентом. Лучшее впечатление производит доклад, в форме пересказа, без зачитывания текста, которым следует пользоваться только для уточнения цифрового материала. Студент должен свободно ориентироваться своей работе.

В выступлении необходимо корректно использовать демонстрационные материалы, которые усиливают доказательность выводов и облегчают восприятие доклада студента. Они оформляются в виде презентации в системе Power Point.

1 Учебно-методическое и информационное обеспечение дисциплины

8.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

8.1.1. Перечень основной литературы:

Васильев, В. В. Информационные технологии в библиотечном деле : учеб.-метод. пособие / Отв. ред. серии О. Р. Бородин. – М. : Либерей-Бибинформ, 2013. – 368 с.

Ларсон, Б. Разработка бизнес-аналитики в Microsoft SQL Server. – СПб. – М. – Н. Новгород : Питер, 2011. – 688 с.

Макин, Дж. К. Проектирование серверной инфраструктуры баз данных Microsoft SQL SERVER : учебный курс : пер. с англ. / Дж. К. Макин, Майк Хотек. – М. : Русская редакция, 2012. – 560 с.

8.1.2. Перечень дополнительной литературы:

- Избачков, Ю. С. Информационные системы: учебник / Ю. С. Избачков, В. Н. Петров. - 3-е изд. - СПб.: Питер, 2011. - 656 с.
- Томас, О. Оптимизация и администрирование баз данных Microsoft SQL Server 2008 : официальное пособие для самоподготовки : пер. с англ. : учебный курс Microsoft. – М. : Русская редакция, 2012. – 601 с.
- Основы проектирования и разработки реляционных баз данных : (Спец. 075200 – Компьютерная безопасность) : учеб. пособие / авт.-сост.: О. М. Лепешкин, Д. Л. Осипов ; Федеральное агентство по образованию, Ставроп. гос. ун-т. – Ставрополь : Изд-во СГУ, 2010. – 203 с.

8.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине:

- Методические указания по выполнению лабораторных работ по дисциплине «Управление данными».
- Методические указания по выполнению контрольной работы по дисциплине «Управление данными».
- Методические рекомендации для студентов по организации самостоятельной работы по дисциплине «Управление данными».
- Методические указания по выполнению курсового проекта по дисциплине «Управление данными».

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины:

- <http://www.intuit.ru> – сайт дистанционного образования в области информационных технологий
- http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=5196 – ЭБС издательства «Лань». Соколова Ю.С., Жулева С.Ю. Разработка приложений в среде Delphi. В 2 частях.
- <http://www.intuit.ru/> – сайт дистанционного образования в области информационных технологий
- <https://msdn.microsoft.com/ru-ru/library/bb545450.aspx> – сайт Библиотека Microsoft SQL Server - MSDN

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023