

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Б1.В.07 Основы программирования и конфигурирования в корпоративных
информационных системах (I часть)**

**Направление подготовки (специальность) 09.03.01 Информатика и вычислительная
техника**

**Профиль образовательной программы Автоматизированные системы обработки
информации и управления**

Форма обучения заочная

СОДЕРЖАНИЕ

1 Конспект лекций	3
1.1 Лекция № 1 Создание новой информационной базы данных.	3
1.2 Лекция № 2 Классификация объектов конфигурации.....	4
1.3 Лекция № 3 Введение в язык запросов	9
2. Методические материалы по выполнению лабораторных работ	17
2.1 Лабораторная работа № ЛР-1 Создание новой информационной базы.....	17
2.2 Лабораторная работа № ЛР- 2 Классификация объектов конфигурации	18
2.3 Лабораторная работа № ЛР- 3 Введение в язык запросов	18
2.4 Лабораторная работа № ЛР- 4 Роли и права пользователей	18

1. КОНСПЕКТ ЛЕКЦИЙ

1.1. Лекция № 1 (2 часа)

Тема: «Создание новой информационной базы данных»

1.1.1. Вопросы лекции:

1. Линейные, иерархические и подчиненные справочники.
2. Предопределенные элементы.

1.1.2. Краткое содержание вопросов:

1. Линейные, иерархические и подчиненные справочники.

Список элементов справочника в системе 1С:Предприятие может быть многоуровневым. В этом случае все строки справочника будут разделяться на 2 вида: «просто» элементы справочника и группы справочника. Группы позволяют переходить на нижележащие уровни многоуровневого справочника. Использование многоуровневых справочников позволяет организовать ввод информации в справочник с нужной степенью детализации. Элементы и группы элементов в многоуровневом справочнике можно переносить из одной группы в другую.

Между справочниками может быть установлено отношение подчиненности. В терминах реляционных баз данных, между таблицами устанавливается связь “один-ко-многим”. В этом случае каждый элемент подчиненного справочника будет связан с одним из элементов справочника-владельца. Иногда можно сказать, что элементы одного справочника принадлежат элементам другого.

Например, в системе может быть справочник НалоговыеЛьготы. Тогда его можно сделать подчиненным справочнику Сотрудники. Это означает, что “Сотрудник владеет налоговыми льготами”.

2. Предопределенные элементы.

Для некоторых объектов метаданных в платформе "1С:Предприятие 8" есть возможность задавать в конфигурации предопределенные элементы, для которых в информационной базе будут автоматически создаваться объекты с заданными значениями. В данной статье рассматриваются некоторые особенности их реализации и работы с ними.

Объекты данных обладают свойством ИмяПредопределенныхДанных, с помощью которого можно управлять их связью с метаданными. Объекты, у которых данное свойство заполнено, являются предопределенными.

Данное свойство может принимать следующие значения:

Имя предопределенного элемента из метаданных.

Пустое значение.

Специальное имя, для идентификации проблемных объектов. Специальное имя начинается с символа #. Его можно установить только в режиме загрузки обмена данными. Режим загрузки нужно включать до присвоения специального имени предопределенного.

Если объекту данных установить пустое значение свойства, то он станет обычным, не предопределенным, объектом.

Если объекту установить имя предопределенного элемента из метаданных, то он станет предопределенным элементом, связанным с метаданными. В пределах одной области информационной базы допустимо использование только одного объекта данных, связанных с конкретным предопределенным элементом метаданных.

В режиме загрузки уникальность предопределенного элемента в пределах области информационной базы не проверяется.

Специальное имя предопределенного является особенностью, которую следует избегать. Такое имя предопределенных данных характерно для удаленных из метаданных предопределенных элементов, при отключенном автоматическом обновлении предопределенных.

Предопределенные элементы можно редактировать, удалять, помечать на удаление. Имеются специальные права для управления ограничениями на удаление предопределенных.

1.2 Лекция № 2 (2 часа)

Тема: «Классификация объектов конфигурации».

1.2.1. Вопросы лекции:

1. Прикладные и подчиненные объекты.
2. Концепция системы.
3. Типы данных.
4. Универсальные коллекции значений.
5. Встроенный язык системы.

1.2.2. Краткое содержание вопросов:

1. Прикладные и подчиненные объекты.

Общие объекты играют вспомогательную, с точки зрения разработки прикладного решения роль. Результат их работы интересен скорее программисту, чем пользователю, хотя нельзя сказать, что эти объекты не участвуют в работе пользователя или неважны для

целей решения прикладной задачи. Они - важная часть конфигурации. Дадим краткую характеристику этих объектов.

Подсистемы. Позволяют структурировать конфигурацию по видам решаемых с ее помощью задач. Например, если конфигурация решает задачи бухгалтерского учета и расчета зарплаты, логично будет создать две подсистемы. При создании новых объектов можно "привязывать" их к подсистемам в зависимости от их предназначения. Использование подсистем облегчает работу с большими конфигурациями.

Общие модули. Содержат процедуры и функции, доступные из других объектов. В этих модулях нельзя определять переменные.

Параметры сеанса. Они нужны, в основном, для настройки прав доступа к данным. Эти параметры устанавливаются при старте системы. Например, в этот параметр может быть записано имя пользователя, вошедшего в систему.

Роли. Предназначены для настройки полномочий различных групп пользователей по доступу к данным. После создания и настройки роли ее можно назначить пользователю (для настройки списка пользователей применяют команду меню Администрирование > Пользователи).

Планы обмена. Определяют правила обмена данными между информационными базами. Например, этот механизм можно использовать для обмена данными между головной организацией и филиалами или между подразделениями организации.

Критерии отбора. Позволяют создавать правила отбора, которые пользователь может применять при работе со списками данных в режиме 1С:Предприятие.

Подписки на события. С помощью подписок на события можно назначать обработчики для событий объектов. Например, обработчик может быть вызван при копировании документа.

Регламентные задания. Регламентные задания позволяют выполнять различные операции по расписанию.

Общие формы. Эти формы могут быть использованы различными объектами для решения схожих задач.

Интерфейсы. Хранит пользовательские интерфейсы (главное меню, набор панелей инструментов), которые можно назначать различным пользователям. Каждый интерфейс можно оснастить лишь теми командами, которые нужны определенному пользователю.

Общие макеты. Хранит общие макеты (шаблоны печатных форм), которые доступны другим объектам конфигурации для формирования печатных вариантов электронных материалов.

Общие картинки. Хранит общедоступные картинки. Их можно, например, использовать в формах при настройке изображений на кнопках и в других случаях.

Пакеты XDTO. XML Data Transfer Object (XDTO) - это универсальный способ представления данных, который предназначен для организации взаимодействия с внешними программными системами. В частности, с другими конфигурациями 1С:Предприятия 8, с другими информационными системами.

WEB-сервисы. Здесь можно создать и настроить WEB-сервисы, которые обеспечивают экспорт функциональности конфигурации.

WS-ссылки. Содержит статические ссылки на опубликованные WEB-сервисы. Позволяет использовать функциональность этих сервисов.

Стили. Содержит наборы настроек форматирования, которые можно применить в ходе разработки. Например - с помощью механизма стилей можно быстро настроить форму заранее заданным образом.

Языки. Позволяют задавать различные языки, на которых можно будет настраивать интерфейс программы. Если в системе задано несколько языков, при настройке объекта появляется возможность задавать некоторые параметры на разных языках. В итоге это приводит к тому, что интерфейс системы может существовать для различных языков.

2. Концепция системы.

Объект – это элемент реального мира. Объектом может быть все что угодно: компьютер, облако, кошка... Но нас интересуют только те объекты, которые имеются в офисной деятельности предприятия: документы, отчеты, справочники... Таким образом, любая конфигурация системы «1С:Предприятие» представляется как совокупность объектов, взаимосвязанных между собой, т.е. конфигурация объектов. Например, документ «Приказ о приеме на работу», справочник «Сотрудники», отчет «Сотрудники, принятые за период».

Под объектом конфигурации в системе «1С:Предприятие» понимается формальное описание группы понятий (предметной области, средств взаимодействия пользователя с системой) со сходными характеристиками и одинаковым предназначением.

Например. Объект конфигурации «Справочник» в системе «1С:Предприятие» предназначен для ведения списков однородных элементов данных — справочников, картотек, нормативных сборников и тому подобное. Использование объектов конфигурации этого типа позволяет организовать ведение любых справочников, необходимых для автоматизации деятельности предприятия.

Каждый объект конфигурации обладает уникальным набором свойств. Этот набор описан на уровне системы и не может быть изменен в процессе настройки конфигурации

задачи. Набор свойств объекта конфигурации определяется, в основном, его назначением в системе «1С:Предприятие».

Главным свойством любого объекта конфигурации является его имя – краткое наименование объекта конфигурации. При создании нового объекта конфигурации ему автоматически присваивается условное имя, состоящее из слова, определяемого по виду объекта, и цифры (например, при создании реквизита создается реквизит с именем «Реквизит1», при создании документа — «Документ1», и т.д.). Это имя можно изменить в процессе редактирования свойств объекта конфигурации, при этом система отслеживает уникальность имен. Имя объекта конфигурации не может пустым.

Некоторые свойства из всего набора свойств, присущих объекту конфигурации, доступны для редактирования и могут быть так или иначе изменены в процессе конфигурирования системы «1С:Предприятие». Характер изменений и их пределы также задаются на уровне системы. Целенаправленным изменением свойств объекта конфигурации специалист, осуществляющий конфигурирование системы, может добиться требуемого поведения объекта при работе системы. Однако такие изменения не затрагивают сущности объекта и не позволяют добиться от него действий, не свойственных объектам данного типа.

Например. Объект конфигурации «Константа» в системе «1С:Предприятие» предназначен для хранения информации, которая не изменяется во времени или изменяется очень редко. Простым примером константы может служить название предприятия: оно, как правило, не меняется в процессе деятельности предприятия.

Константа обладает следующим набором редактируемых свойств:

- имя константы;
- синоним;
- комментарий;
- подсистемы;
- тип данных.

Помимо набора свойств объекты имеют определенные методы (процедуры и функции). Например, вы можете выбрать всех сотрудников с помощью метода справочника Выбрать(), или можно сохранить документ методом Записать().

3. Типы данных.

Любые данные, т.е. константы, переменные, значения функций или выражения, в Турбо Паскале характеризуются своими типами. Тип определяет множество допустимых значений, которые может тот или иной объект, а также множество допустимых операций,

которые применимы к нему. Кроме того, тип определяет формат внутреннего представления данных в памяти ПК.

К простым типам относятся порядковые и вещественные типы.

Порядковые типы отличаются тем, что каждый из них имеет конечное число возможных значений. Эти значения можно определённым образом упорядочить (отсюда – название типов) и, следовательно, с каждым из них можно сопоставить некоторое целое число – порядковый номер значения.

Вещественные типы тоже имеют конечное число значений, которое определяется форматом внутреннего представления вещественного числа. Однако количество возможных значений вещественных типов настолько велико, что сопоставить с каждым из них целое число (его номер) не представляется возможным.

4. Универсальные коллекции значений.

Коллекция значений – это некий контейнер, в котором может содержаться обычно любое количество элементов.

При этом каких-либо жестких ограничений на тип данных зачастую не накладывается. В универсальную коллекцию можно добавлять значения. Все значения в коллекции можно обойти. Используются эти коллекции в основном для какой-либо обработки в алгоритмах. Т.е. это некие динамические структуры, которые существуют на время работы алгоритма. Важно понимать, что коллекции не хранятся в базе данных (о типе данных Хранилище значений, которое может сохранять практически любой тип данных, речь пока не идет).

Существуют различные виды универсальных коллекций: Массив, Структура, Соответствие, Фиксированный массив, Таблица значений, Табличная часть и т.д. Но у всех коллекций есть схожесть поведения. Коллекция может создаваться в результате работы какой-либо функции (функция возвращает в качестве значения универсальную коллекцию).

Можно получить новую коллекцию вручную, обратившись к конструктору и создав экземпляр класса. Например: НашМассив = Новый Массив; Конструкторы для многих универсальных коллекций являются параметризованными. Так, в конструкторе для Массива можно указать количество элементов в соответствующих измерениях. Т.е. можно сразу же объявлять многомерные Массивы.

5. Встроенный язык системы.

Строенный язык является важной частью технологической платформы 1С:Предприятия 8, поскольку позволяет разработчику описывать собственные алгоритмы функционирования прикладного решения.

Встроенный язык имеет много общих черт с другими языками, такими как Pascal, Java Script, Basic, что облегчает его освоение начинающими разработчиками. Однако он не является прямым аналогом какого-либо из перечисленных языков.

Вот лишь некоторые, наиболее значимые особенности встроенного языка:

- предварительная компиляция - перед исполнением модули, содержащие текст на встроенном языке, преобразуются во внутренний код;
- кэширование скомпилированных модулей в памяти;
- мягкая типизация - тип переменной определяется типом значения, которое она содержит, и может изменяться в процессе работы;
- отсутствие программного описания объектов конфигурации - разработчик может использовать либо встроенные в платформу объекты, либо объекты, созданные системой в результате визуального конструирования прикладного решения.

1.3 Лекция № 3 (2 часа)

Тема: «Введение в язык запросов».

1.3.1. Вопросы лекции:

1. Источники данных и табличная модель данных.
2. Основы синтаксиса языка запросов.
3. Введение в компоновку данных - предыстория создания и основные возможности механизма.
4. Формирование отчетов с помощью запросов.
5. Конструктор запросов.
6. Доступные поля отчета.

1.3.2. Краткое содержание вопросов:

1. Источники данных и табличная модель данных.

Таблица — это объект БД, который служит для ввода и хранения информации. Таблица состоит из записей (строк), которые составляют информацию, хранящуюся в ней, и полей (столбцов), образующих структуру базы данных.

Для каждого поля задается имя поля, тип данных, перечень свойств, описание.

Имя поля должно быть уникальным, содержать не более 64 символов, включая пробелы, причем имя поля не должно начинаться с пробела. Имена полей могут содержать любые символы, за исключением (.); (!); ('); ([]); ()).

Тип данных определяет вид и диапазон значений, которые могут содержаться в данном поле. Microsoft Access предлагает следующие типы данных.

Текстовой — тип данных, используемый для хранения символьных или числовых данных, не требующих вычислений. Поле данного типа может содержать до 255 символов.

Числовой — тип данных для хранения числовых значений, используемых в математических расчетах. Данные этого типа могут содержаться в 1, 2, 4, 8 байтах в зависимости от значения поля Размер поля.

Поле Memo — специальный тип данных для хранения больших объемов информации, по объему превышающей 255 символов. Это поле может содержать до 65535 символов.

Дата / время — тип данных для хранения календарных дат и текущего времени.

Денежный тип — тип данных для хранения денежных значений, точность представления которых колеблется от 1 до 4 знаков после запятой. Целая часть может содержать до 15 десятичных знаков. Длина поля 8 байт.

Счетчик — специальный тип данных для уникальных натуральных чисел, обладающий свойством автоматического наращивания при каждой новой записи. Длина поля 4 байта.

Логический тип — специальный тип данных, имеющий только два значения, интерпретируемых как Да/Нет, Истина/Ложь, Вкл/Выкл. Длина поля 1 байт.

Поле объекта OLE — содержит ссылку на OLE-объект, например документы Microsoft Word, Microsoft Excel, звук, изображение и другие виды объектов. Ограничено дисковым пространством.

Гиперссылка — специальное поле, предназначенное для хранения гиперссылки, с помощью которой можно ссылаться на произвольный фрагмент данных внутри файла или страницы на том же компьютере, в локальной сети или Интернете. Длина до 2048 символов.

При выборе типа данных для каждого поля открывается соответствующий набор свойств, с помощью которого пользователь определяет вид и функциональные характеристики данного поля.

Наиболее важными являются следующие свойства.

Размер поля — определяет максимальную длину данных, которые могут содержаться в данном поле.

Формат поля — определяет способ формирования данных в данном поле таблицы.

Число десятичных знаков — определяет количество знаков после запятой для полей числового и денежного типа.

Маска ввода — определяет форму ввода данных в поле и облегчает контроль за вводимыми символами.

Подпись — задает названия полей таблицы, которые выводятся в различных режимах.

Значение по умолчанию — указывает значение, автоматически вводимое в поле при создании новой записи.

Условие назначения — ограничение, накладываемое на вводимые в данное поле данные.

Сообщение об ошибке — это текст сообщения, которое будет выводиться в случае нарушения условия назначения.

Обязательное поле — определяет, является ли ввод данных в это поле обязательным.

Индексированное поле — определяет, является ли данное поле индексированным. Индекс ускоряет все операции, связанные с сортировкой, группировкой или поиском записей по значению.

Желательно, чтобы каждая таблица имела ключевое поле. Ключ однозначно определяет каждую запись в таблице; повторяющиеся значения ключа не допускаются. Связываться могут только таблицы, имеющие ключевые поля.

Для определения ключа выделяется, как правило, одно поле, и на панели инструментов нажимается кнопка Ключ с изображением ключа (логотип Access). Однако в ситуации, когда данные одного поля не могут быть уникальными для каждой записи, можно назначить ключевыми два или более полей.

Ключевые поля таблицы индексируются автоматически.

При вводе данных в поле осуществляется автоматическая проверка типа полей данных. Кроме того, можно установить дополнительные правила проверки вводимых данных в свойстве поля Условие на значение, которое позволяет обеспечить ввод корректных данных в поле записи. Проверка значения осуществляется при переходе к другому полю или другой записи. Если введенное значение не соответствует заданному условию, то на экран выводится сообщение об ошибке.

2. Основы синтаксиса языка запросов.

Программа? на языке SQL представляет собой простую линейную последовательность операторов языка SQL. Язык SQL в своем ?чистом? виде операторов управления порядком выполнения запросов к БД (типа циклов, ветвлений, переходов) не имеет.

Операторы языка SQL строятся с применением:

зарезервированных ключевых слов;

идентификаторов (имен) таблиц и столбцов таблиц;

логических, арифметических и строковых выражений, используемых для формирования критериев поиска информации в БД и для вычисления значений ячеек результирующих таблиц;

идентификаторов (имен) операций и функций, используемых в выражениях.

Все ключевые слова, имена функций и, как правило, имена таблиц и столбцов представляются 7-мибитными символами кодировки ASCII (иначе говоря - латинскими буквами).

В языке SQL не делается различия между прописными (большими) и строчными (маленькими) буквами, т.е., например, строки ?SELECT?, ?Select?, ?select? представляют собой одно и то же ключевое слово.

Для конструирования имен таблиц и их столбцов допустимо использовать буквы, цифры и знак _? (подчеркивание), но первым символом имени обязательно должна быть буква.

Запрещено использование ключевых слов и имен функций в качестве идентификаторов таблиц и имен столбцов. Полный список ключевых слов и имен функций (а он весьма обширен) можно найти в документации на конкретную СУБД.

Оператор начинается с ключевого слова-глагола (например, ?CREATE? - создать, ?UPDATE? - обновить, ?SELECT? - выбрать и т.п.) и заканчивается знаком ?;? (точка с запятой). Оператор записывается в свободном формате и может занимать несколько строк. Допустимыми разделителями лексических единиц в операторе являются:

один или несколько пробелов,

один или несколько символов табуляции,

один или несколько символов ?новая строка?.

При описании операторов языка SQL в учебном пособии используются следующие соглашения.

Прописными (большими) буквами (например, SELECT, FROM, WHERE) набраны зарезервированные слова.

Курсивом (например, имя_табл, сложн_условие) набраны переменные (нетерминальные символы), подлежащие замене в реальном операторе конструкцией из терминальных символов (идентификаторов, знаков операций, имен функций и т.п.).

В квадратные скобки ?[...]? заключается необязательная часть оператора, которую можно опустить при создании реального оператора (сами квадратные скобки в текст оператора не включаются).

Вертикальная черта (`?|?`) означает возможность выбора (?или?) из двух или нескольких вариантов синтаксической конструкции (сама вертикальная черта в текст оператора не включается). Подчеркнутый вариант (например, в `?[ALL | DISTINCT]?`) является умолчательным.

Последовательность символов `? , ...?` обозначает возможность повторения произвольное количество раз (в том числе и нулевое) предшествующей запятой конструкции. Символ `? ,?` включается в реальный оператор в качестве разделителя перед каждым повторением конструкции.

К сожалению, разработчики реальных СУБД неаккуратно обращаются с требованиями стандартов языка SQL в части комментариев. Поэтому комментарии при использовании в различных СУБД в текстах ?программ? на языке SQL могут помечаться следующими способами:

- от двойного минуса (`?--?`) до конца строки;
- от символа `?#?` до конца строки;
- между последовательностями `?/*?` и `?*?/` (стиль комментариев языка СИ).

3. Введение в компоновку данных - предыстория создания и основные возможности механизма.

Система компоновки данных (СКД) предназначена для получения и манипуляции информацией, имеющей сложную структуру, с возможностью дальнейшего ее представления в виде произвольного набора таблиц и диаграмм. Применение механизма компоновки данных позволяет дать целостное описание отчета, используя, в первую очередь, только визуальные средства, и реализовать следующие возможности:

- создание отчета без программирования;
- использование автоматически генерируемых форм просмотра и настройки отчета;
- разбиение исполнения отчета на этапы;
- исполнение отдельных этапов построения отчета на различных компьютерах;
- независимое использование отдельных частей системы компоновки данных;
- программное управление процессом выполнения отчета.

Основная задача данного курса познакомить слушателей с устройством системы компоновки данных платформы "1С:Предприятия 8". Дать им практические навыки по использованию данного механизма в целях отображения сводной информации различной степени сложности, в первую очередь при формировании отчетов.

4. Формирование отчетов с помощью запросов.

Информацию для создания какого-либо отчета чаще всего приходится искать в документах. Данные из документов определенным образом агрегируются и группируются. Так получаются данные для окончательного отчета. Одним из инструментов быстрого получения информации из документов среды 1С:Предприятие является построение запросов.

В системе 1С:Предприятие существует специальный агрегатный тип данных - Запрос. Запросы формулируются на специальном языке запросов. В тексте запроса присутствуют следующие секции:

объявление переменных запроса (ссылки на атрибуты или реквизиты объектов), например,

Количество=Документ.ПоступлениеМатериалов.Количество

или

Поставщик=Документ.ПоступлениеМатериалов.Поставщик

описание функций агрегирования, значения которых определяются на основании переменных запроса, например,

Функция Сумма1=Сумма(Количество)

описание группировок запроса (по каким переменным необходимо сгруппировать данные и как их упорядочить), например,

Группировка Поставщик упорядочить по Поставщик.Наименование

Текст запроса удобно сохранить в строковой переменной. После формирования текста запроса следует его выполнить. Для этого требуется создать объект агрегатного типа данных Запрос (например, МойЗапрос) и вызвать для него метод Выполнить(ТекстЗапроса). Эта функция вернет 0, если в запросе была ошибка, в противном случае будет возвращена 1.

Далее требуется осуществить перебор строк полученной выборки. Перебор осуществляется с помощью функции Группировка(), параметром которой является НомерГруппировки или ИмяГруппировки, заданное при формулировке запроса. Функция Группировка() возвращает 1, если получена новая строка с данными следующей группы. Получить значение по этой группе можно с помощью метода ЗначениеУпорядочивания(Группировка, Упорядочивание) типа данных Запрос. Первый параметр функции задает номер используемой группировки или ее имя, а второй параметр - номер или имя признака сортировки, заданного в описании этой группировки (сортировка может производиться, к примеру, по нескольким реквизитам объекта, который является основанием группировки).

При обработке результатов запроса к значениям функций агрегирования следует обращаться без указания скобок. Например, Запрос.Функция1.

Полезной также является процедура, которая позволяет выгружать результат выполнения запроса в таблицу значений и затем анализировать эти данные с помощью возможностей агрегатного типа данных ТаблицаЗначений:

```
T=СоздатьОбъект("ТаблицаЗначений");
```

```
Запрос=СоздатьОбъект("Запрос");
```

```
...
```

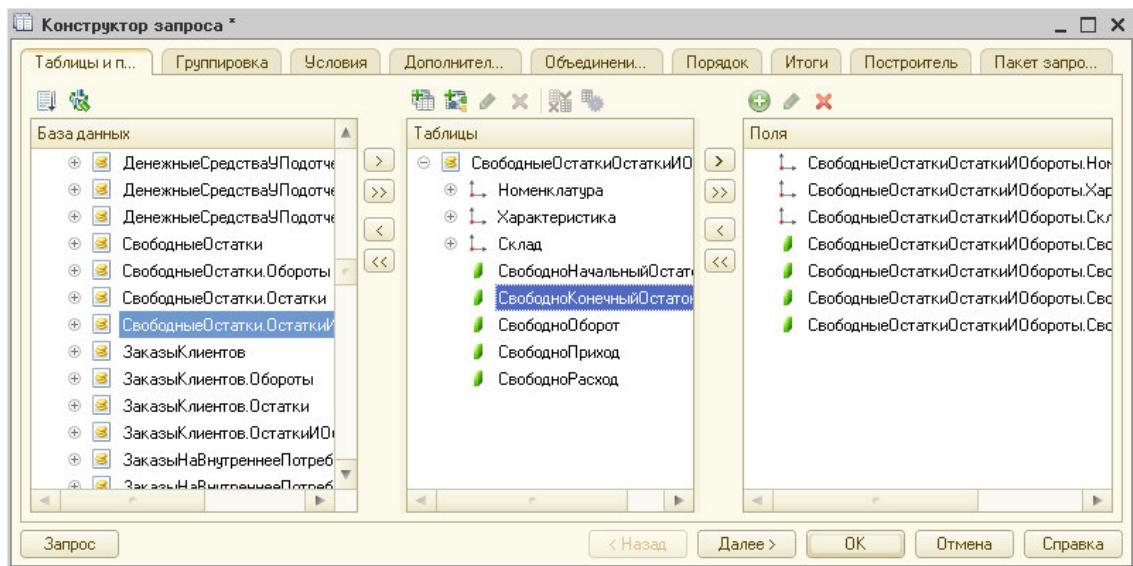
```
Запрос.Выгрузить(T);
```

Для создания запросов в отчете или другом элементе конфигурации можно воспользоваться конструктором запроса. Например, откроем форму некоторого отчета, выбираем меню Конструкторы-Запрос. Нам будут предложены диалоги, в которых последовательно можно будет задать, будет ли запрос иметь период, задать имена переменных, функций агрегирования, поля группировки. В последнем диалоге нужно проследить, чтобы были выбраны флаги Генерировать процедуру и Генерировать таблицу. Таким образом, будет сгенерирована не только процедура обработки запроса, но и все секции таблицы вывода информации для этого отчета. Далее сгенерированную таблицу и процедуру можно редактировать произвольным образом.

5. Конструктор запросов.

Конструктор запроса - это один из инструментов разработки. Он позволяет составить текст запроса на языке запросов исключительно визуальными средствами.

С помощью кнопок **Далее** и **Назад** можно перемещаться по закладкам конструктора и указывать, какие данные должны присутствовать в результате запроса, как они связаны, сгруппированы, какие итоги следует рассчитать, работать с временными таблицами, редактировать пакет запросов:



Результатом работы конструктора будет являться синтаксически правильный текст запроса. Таким образом, разработчик может составить работоспособный запрос, даже не владея синтаксисом языка запросов - необходимые синтаксические конструкции конструктор сгенерирует автоматически. Готовый текст запроса может быть сразу же вставлен в текст модуля или скопирован в буфер обмена.

```

Запрос
ВЫБРАТЬ РАЗЛИЧНЫЕ
    ПартнерыКонтактнаяИнформация.Ссылка КАК Партнер,
    ПартнерыКонтактнаяИнформация.Ссылка.Наименование,
    ПартнерыКонтактнаяИнформация.АдресЭП КАК АдресЭП
ИЗ
    Справочник.Партнеры.КонтактнаяИнформация КАК ПартнерыКонтактнаяИнформация
ГДЕ
    ПартнерыКонтактнаяИнформация.АдресЭП ПОДОБНО &СтрокаПоиска

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ
    КонтактныеЛицаПартнеровКонтактнаяИнформация.Ссылка.Владелец,
    КонтактныеЛицаПартнеровКонтактнаяИнформация.Ссылка.Владелец.Наименование,
    КонтактныеЛицаПартнеровКонтактнаяИнформация.АдресЭП
ИЗ
    Справочник.КонтактныеЛицаПартнеров.КонтактнаяИнформация КАК КонтактныеЛицаПартнеровКонтактна
ГДЕ
    КонтактныеЛицаПартнеровКонтактнаяИнформация.АдресЭП ПОДОБНО &СтрокаПоиска

УПОРЯДОЧИТЬ ПО
    Партнер,
    АдресЭП
ИТОГИ ПО
    Партнер

```

Кроме этого конструктор запросов позволяет редактировать уже имеющийся в программе текст запроса. Для этого достаточно установить курсор внутри существующего текста запроса и вызвать конструктор. Имеющийся текст запроса будет проанализирован и представлен в конструкторе в виде соответствующих выбранных полей базы данных и набора заданных связей, группировок, условий и т.д.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1 Лабораторная работа №1 (2 часа).

Тема: «Создание новой информационной базы».

2.1.1. Краткое описание проводимого занятия:

Задания для проведения текущего контроля успеваемости

1. С какими СУБД работает система «1С: Предприятие 8.2»?
 - 1) Microsoft SQL Server, File DBMS, Fox Pro, Oracle Database
 - 2) Microsoft SQL Server, PostgreSQL, MS Access
 - 3) Microsoft SQL Server, PostgreSQL, IBM DB2
 - 4) Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database
 - 5) Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database, File DBMS
2. Какого вида клиентского приложения не существует в системе «1С: Предприятие 8.2»?
 - 1) отладочный клиент
 - 2) толстый клиент
 - 3) тонкий клиент
 - 4) веб-клиент
 - 5) не существует 2 и 3 вариантов

3. Для каких целей может использоваться «Толстый клиент»?
 - 1) использование прикладного решения
 - 2) отладка прикладного решения
 - 3) разработка прикладного решения
 - 4) верны ответы 1 и 2
 - 5) верны все варианты

4. Для каких целей может использоваться «Тонкий клиент»?
 - 1) использование прикладного решения
 - 2) отладка прикладного решения
 - 3) разработка прикладного решения
 - 4) верны ответы 1 и 2
 - 5) верны все варианты

2.2 Лабораторная работа №2 (2 часа).

Тема: «Классификация объектов конфигурации».

2.2.1 Краткое описание проводимого занятия:

Задания для проведения текущего контроля успеваемости

1. Произвольная классификация объектов конфигурации осуществляется с помощью:
 - 1) свойства Комментарий у объекта конфигурации
 - 2) создание реквизита «Классификация»
 - 3) с помощью объекта конфигурации Подсистема
 - 4) верны варианты 1 и 2
 - 5) верны все варианты
2. Какой из вариантов лучше всего предложить пользователю, чтобы привести учет, контролируемый последовательностью, в актуальное состояние?

- 1) воспользоваться режимом восстановления последовательности из диалога «Операции» / «Проведение документов» / «Восстановление последовательностей»
 - 2) перепровести в хронологической последовательности все документы, принадлежащие условию
 - 3) перепровести в хронологической последовательности все документы, принадлежащие последовательности
 - 4) перепровести в хронологической последовательности все документы, принадлежащие последовательности по формуле:
Последовательности.ИмяПоследовательности.ПолучитьГраницуПоследовательности().Ссылка
3. Определите отличие между вводом документа копированием и вводом «на основании»:
- 1) копирование – создание нового объекта копированием, а ввод «на основании» – создает объект, зависящий от «объекта-основания»
 - 2) между этими операциями нет никакого существенного различия
 - 3) при копировании документ создается и записывается в ИБ, а при вводе – изменить скопированные данные
 - 4) верны ответы 2 и 3 – нет различия и документ создается, а также записывается в ИБ путем копирования

2.3 Лабораторная работа №3 (2 часа).

Тема: «Введение в язык запросов».

2.3.1 Краткое описание проводимого занятия:

Задания для проведения текущего контроля успеваемости

1. Использование конструктора запросов позволяет:
 - 1) сформировать текст нового запроса в конструкторе
 - 2) сформировать программный код на создание объекта Запрос, текста запроса и результат
 - 3) отредактировать текст имеющегося запроса в конструкторе
 - 4) верны предложенные варианты ответов – один и три
 - 5) все вышеперечисленные ответы являются верными
2. Использование конструктора запросов с обработкой результата позволяет:
 - 1) сформировать текст нового запроса в конструкторе
 - 2) сформировать программный код на создание объекта Запрос, текст запроса, результат обработки
 - 3) отредактировать текст имеющегося запроса в конструкторе
 - 4) верны предложенные варианты ответов – один и три
 - 5) все вышеперечисленные ответы являются верными
3. При сохранении текста запроса, открытого повторно с помощью конструктора запроса, без внесения в этот текст изменений:
 - 1) весь первоначальный текст остается без изменений
 - 2) из первоначального текста будут удалены только комментарии
 - 3) в первоначальном тексте останутся комментарии и удалены конструкции языка запросов построителя отчетов
 - 4) из первоначального текста будут удалены и комментарии, и конструкции языка запросов построителя отчетов

2.4 Лабораторная работа №4 (2 часа).

Тема: «Роли и права пользователей».

2.10.1 Краткое описание проводимого занятия:

Задания для проведения текущего контроля успеваемости

1. Роль, как объект конфигурации отражает:

- 1) статус и сферу ответственности ее носителя как сотрудника компании
- 2) совокупность доступных определенному пользователю объектов в среде «1С: Предприятие»
- 3) совокупность прав действий в среде «1С: Предприятие»
- 4) набор параметров пользователя

2. Для запрета режима непосредственного удаления объектов пользователем необходимо:

- 1) в свойствах конфигурации снять флаг «Разрешить непосредственное удаление объектов»
- 2) при определении роли для соответствующих объектов снять отметку у флага «Удаление»
- 3) при определении роли для объектов снять отметку флага «Интерактивное удаление»
- 4) в свойствах роли снять отметку у флага «Разрешить удаление объектов»
- 5) в свойствах роли снять отметку у флага «Разрешить интерактивное удаление объектов»

3. Режим временной блокировки предназначен для:

- 1) блокировки подключений пользователей к информационной базе
- 2) предотвращения несанкционированного использования системы «1С: Предприятие» в отсутствие пользователя
- 3) для временной блокировки объекта информационной базы для использования другими пользователями
- 4) блокировки активных пользователей с целью высвобождения ресурсов платформы