**Разработка документа под ключ в системе БИТ.ВУЗ.**

Требуется разработать документ, позволяющий привязывать в разрезе истории чтение дисциплин кафедрами в разрезе формы обучения на факультетах. Каждый документ для факультета с указанной даты задает исчерпывающий список дисциплина+кафедра+форма обучения ,которые действуют на факультете.

**Для выполнения задачи:**

1. Разворачиваем демо базу БИТ.ВУЗ последнего релиза.

**Комментарий**: нужно решать задачу так, как если бы нам заранее неизвестна периодичность регистра, т.е. мы можем ее поменять на любую другую и все должно работать.

**Постановка задачи:**

1. Дан каркас документа «ДисциплиныКафедрПоФакультетам» и регистры (см. хранилище БИТ.ВУЗ). Обеспечить формирование и запись движений документа.
2. Создать роли для просмотра, изменения с РЛС по виду доступа Факультеты разрабатываемого документа. Запрещено в эти роли включать запуск к каким-либо клиентам (тонким, толстым…). Роль должна включать доступ к РС, куда записываются движения.

Чтобы активировать роль необходимо сначала в предприятии запустить обработку Обновление вспомогательных данных (обновить), после этого в предприятии можно роль использовать в профиле групп доступа. Ограничения РЛС задаются в группе доступа, активируются в профиле групп доступа. Пользователю назначают через группы доступа в предприятии.

1. Разместить документ и его регистры в интерфейсе Расписания. Регистры не показывать в этой подсистеме.
2. Документ должен работать в веб-клиенте. Запрещено перебирать строки ТЧ или осуществлять поиск строк ТЧ на клиенте! ТЧ.НайтиСтроки() на клиенте не должно использоваться, так как приводит к неявным серверным вызовам.
3. Документ должен помнить, кто его последний трогал и когда трогал. Пользователь на редактирование не должен иметь доступ к этим полям.
4. Должен блокировать себя от редактирования интерактивно, если пользователь не имеет полных прав и не является ответственным (для образца можно глянуть как это реализовано в др. документах конфигурации). См. документ «Расписание группы».
5. Две табличные части должны быть связаны по ключу и находиться на одном экране. При выделении кафедры надо показывать в связанной ТЧ только ее дисциплины. Надо организовать автоинкрементый счетчик. Вычислять значение счетчика надо ровно 1 раз, далее просто его плюсовать. Ключ не показывать пользователю.
6. Модальных вызовов быть не должно. Нельзя использовать Вопрос, ОткрытьМодально, надо использовать их асинхронные аналоги. Вызов серверных функций в событиях формы, обрабатывающих изменение, удаление, добавление данных запрещен, используйте серверные вызовы асинхронно: через обработчик ожидания. Вызов серверных методов в цикле запрещен!
7. При вводе дисциплин сначала спрашиваем, какой из двух составных типов желаем. Потом обеспечиваем множественный ввод дисциплин в ТЧ, то есть, если я выберу 6 дисциплин, то при нажатии на выбрать в форме выбора дисциплин у меня будет добавлено 6 дисциплин в ТЧ для текущей кафедры. Формочку выбора не создаем, используем типовую. Пример: в документе «Установка видов занятий расписания» множественный ввод видов занятий в ТЧ. Обработать след. ситуацию: пользователь добавляет кафедру копированием имеющейся строки, копируются подчиненные строки, курсор встает на новую строку, пользователь нажимает кнопку Esc – остаются неудаленными строки в подчиненной ТЧ, такого быть не должно.
8. Запрещено выбрать в ТЧ помеченные на удаление объекты справочников. Использовать для этого Параметры выбора у реквизита объекта!
9. Проверки в документе (не выполнять проверки при импорте и попытке установить пометку удаления, все проверки должны выполняться запросом, а не перебором строк ТЧ, вывод сообщений привязывать к номерам строк ТЧ с кафедрами). Проверки при записи документа – предупреждающие, при проведении – запрещающие. Также сообщение должно начинаться со слов «Предупреждение:», если ошибка не запрещающая, иначе должны быть слова «Ошибка: »:
   1. - если есть проведенный документ, который уже для данной кафедры зарегистрировал чтение дисциплины в разрезе указанной форме обучения на другом факультете, но в пределах даты среза (то есть, актуальных на дату среза (Объект.Дата)), то выдаем предупреждающее сообщение при записи и запрещающую при проведении с указанием полного сообщения, вывода номера документа, который уже первый занял эту дисциплину и указание строки ТЧ кафедр нашего документа, где произошла проблема. В выводимом сообщении должен содержаться отсыл к текущему документу. То есть, контролируем, чтобы одновременно действующих документов на определенную дату с иными факультетами, но одинаковыми сочетаниями кафедра+дисциплина+форма обучения не было.
   2. - проверять на дублирование строк в обоих ТЧ в документе, вывод сообщений аналогичен проверке выше, при этом требуется выводить сообщение с указанием номеров дублируемых строк, то есть, если строка 1 пересекается с 2 и 3, то выводить надо для строки 1 лишь сообщение о пересечении со строкой 2 или 3, чтобы не захламлять сообщениями пользователя. Обеспечить работу привязки сообщения к полю, даже если номер строки более 1000.

При выводе сообщений предусмотреть, чтобы привязка к полям ТЧ работала, даже если номер строки более 999. Текст сообщения должен содержать явное указание номера строки, где проблема, дисциплины/кафедры/формы обучения.

1. При копировании кафедры надо копировать подчиненные строки ТЧ дисциплин.
2. В документе нужна кнопка сортировки ТЧ Дисциплины по любому полю ТЧ, кроме ключа и номера строки, настройки сортировки надо хранить в пользовательских настройках средствами платформы. Пример можно посмотреть в обработка «РасстановкаППСВРасписании».
3. В документе нужна кнопка «Заполнить документ по данным движений предыдущего документа» (В пределах факультета). Перед заполнением, если ТЧ не пуста, то требуется выдавать вопрос (немодальный) с предупреждением о том, что ТЧ будет очищена. После выполнения заполнения надо устанавливать сортировку, которую задавал пользователь.
4. Заполнять ответственного текущим пользователем, если он был пуст.
5. Отключить использование в элементе формы с каждой ТЧ АвтоВводНовойСтроки и АвтоВводНезаполненного.
6. Список выбора кафедр формировать по данным предыдущего действующего документа (в пределах факультета), но при этом не отключать возможность выбрать кафедру из формы списка. Всегда, когда

есть ограничения на список выбора чего-либо, необходимо у реквизита/элемента отключать историю при вводе.

1. Разработать макет и ПФ с возможностью печати из формы списка сразу множества документов. См. приложение № 1 в качестве образца. При выводе на печать использовать механизмы БСП в БИТ.ВУЗ (общая форма ПечатьДокументов). Образец использования в ПФ документа «Аттестационная ведомость». Кнопка печати должна быть включена в роли. Заголовок ПФ должен быть по центру страницы (по горизонтали). Проверить, что выделив несколько документов при печати заголовок не прыгает из стороны в сторону, а находится ровно по центру каждой из выводимой ПФ, то есть, цель: выделил 1 документ, распечатал, выдели второй документ распечатал, выделил оба сразу, распечатал, то в каком бы варианте не печатался первый или второй документ его ПФ должна быть неизменной на принтере.
2. При удалении строк обоих ТЧ спрашивать подтверждение. Удалять можно несколько строк ТЧ за раз. При удалении кафедры должны удаляться все связанные с ней дисциплины.
3. Проверки при записи/проведении делаем в модуле объекта перед записью.
4. Все вопросы, задаваемые пользователю должны иметь кнопку Да по умолчанию, а также иметь кнопку Отмена.
5. Печатная форма должна в точности повторять образец, в том числе шрифт и размер шрифта. При печати рекомендуем использовать метод ТабДок.Присоединить().
6. Собственные движения документ не должен никогда учитывать ни в каких из вышеперечисленных задачах.
7. Разработать отчет, который выведет для каждого документа на дату его действия специальности студентов, для каждой специальности вывести дисциплины, формы обучения, кафедры. Если студентов за документом нет, это не значит что не надо выводить дисциплины и кафедры. Также выводить количество студентов, находящихся в допустимом списке состояний (задается параметром отчета пользователем, но уже предзаполнен), для каждого документа. Выводить итог по численности студентов для каждого документа. На уровне документа выводить факультет. Жирным шрифтом выводить те документы, которые действующие на дату отчета, при этом сам признак того, что документ действующий или нет не должен выводиться среди полей отчета. Количество студентов и специальности студентов вычислять не на дату отчета, а на дату действия каждого документа. Студент считается, что принадлежит документу, если факультет студента совпадает с факультетом документа на дату документа и студентов в массиве допустимых состояний (см. БИТ.ОбщегоНазначенияСервер. ПолучитьСписокДопустимыхСостоянийСтудента(ЛОЖЬ)) на дату действия документа. Факультет студента лежит в РС Обучение, состояние в РС СостояниеСтудента. Выводить специальность и код специальности в разных полях отчета. Никаких других лишних полей в отчет и запрос не выводим. При разработке отчета обратить внимание на то, что параметры отчета должны быть обязательными для заполнения, и не требуется нажимать никаких доп. кнопок, чтобы их увидеть. Расположение полей: в первой группировке Документ, Факультет; на второй группировке Специальность, Код,; в детальных записях выводим кафедру, дисциплину, форму обучения - на всех уровнях выводим численность студентов. Кол. студентов считаем запросом, а не средствами СКД. Если возникнет непонимание поведения отчета или будут возвращаться данные, отличающиеся от исполнения запроса в консоли (это свидетельство того, что в результате работы оптимизатора запроса со стороны СКД 1С была произведена им неверная оптимизация, которую следует подавить), то следует воспользоваться консолью СКД (<https://its.1c.ru/db/metod8dev#content:4099:hdoc>), она расположена во Внешних отчетах (сетевой ресурс). Если возникнут проблемы с тем, что количество студентов документа увеличивается на количество дисциплин в документе, то рекомендуется в качестве агрегатной функции СКД использовать функцию ВычислитьВыражениеСГруппировкойМассив (погугли), например Сумма(ВычислитьВыражениеСГруппировкойМассив("Максимум(КолСтудентов)", "Документ"));

Пример поведения отчета: студента зачислили 01.09 на факультет1, студента перевели на факультет2 с 01.10, студента перевели на факультет1 с 01.11, студента отчислили 01.12; документ1 на 01.09 на факультет1, документ2 на факультет2 на 01.09, документ3 на 01.10 на факультет1, документ4 на факультет2 на 01.10, документ5 на 01.11 на факультет1, документ6 на факультет2 на 01.11, документ7 на 01.12 на факультет1, документ8 на факультет2 на 01.12. Отчет должен вывести:

|  |  |  |
| --- | --- | --- |
| **Документ** | **Факультет** | **Кол. обучающихся** |
| 1 | 1 | 1 |
| 2 | 2 | 0 |
| 3 | 1 | 0 |
| 4 | 2 | 1 |
| 5 | 1 | 1 |
| 6 | 2 | 0 |
| 7 | 1 | 0 |
| 8 | 2 | 0 |

**Приложение № 1.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **[Имя ВУЗа]** | | | | | | |
| **Матрица покрытия дисциплин кафедрами факультета [ИмяФакультета]** | | | | | | |
|  |  |  |  |  |  |  |
| **№** | **Кафедра/Форма обучения** | **Дисциплина1** | **Дисциплина2** | **Дисциплина3** | **Дисциплина4** | **Дисциплина5** |
| **1** | **Кафедра1** | Х |  | Х |  | Х |
| 1.1 | Очная | Х |  | Х |  |  |
| 1.2 | Очно-заочная |  |  | Х |  | Х |
| **2** | **Кафедра2** | Х | Х |  | Х |  |
| 2.1 | Очная | Х | Х |  | Х |  |