



Динамические Web-страницы с JavaServer Faces

Содержание

Отображение динамических сведений на Web-страницах с JavaServer Faces . 1

Отображение динамических сведений на Web-страницах с JavaServer Faces	1
Модуль 1. Создание Web-страниц с соединениями с данными.	3
Урок 1.1. Импортирование необходимых ресурсов и настройка целевого сервера	4
Урок 1.2. Подключение к базе данных и отображение данных на Web-странице	6
Урок 1.3. Тестирование Web-сайта	12
Урок 1.4. Создание записи для отображения и обновления базы данных	14
Урок 1.5. Программирование кнопки отправки	17

Урок 1.6. Создание страницы обновления.	20
Урок 1.7. Фильтрация списка реляционных записей	25
Модуль 1 - итоги	29
Модуль 2. Добавление дополнительных функций	31
Урок 2.1. Форматирование таблицы данных	34
Урок 2.2. Применение компонента загрузки файлов	40
Урок 2.3. Применение правил навигации	43
Урок 2.4. Применение автоматической генерации ключей	45
Модуль 2: итоги	48
Отображение динамических сведений на Web-страницах с JavaServer Faces	48

Отображение динамических сведений на Web-страницах с JavaServer Faces

Это учебник по разработке динамического Web-сайта, работающего как раздел рекламы и объявлений в газете. Web-сайт - это полное Web-приложение J2EE, использующее компоненты JavaServer Faces (JSF) и Service Data Objects (SDO). Web-приложение применяет технологию JSF для создания динамических Web-страниц, содержащих ссылки на базу данных с объявлениями.

Цели обучения

Этот учебник разбит на два модуля, каждый из которых преследует свои цели обучения. Вы можете изучить любой из них или оба модуля. В каждом модуле вы должны выполнить упражнения, чтобы Web-сайт работал правильно.

Модуль 1. Создание Web-страниц с соединениями с данными

Благодаря работе с данным модулем вы узнаете, как настроить соединение с базой данных и использовать информацию из нее на Web-страницах. В результате работы с модулем вы научитесь:

- Подключать Web-страницы к базе данных.
- Создавать страницы, способные отображать, создавать, изменять и удалять записи из базы данных.
- Создавать базовую страницу результатов поиска.
- Отправлять данные с одной страницы на другую.

Модуль 2. Добавление дополнительных функций

В этом модуле описаны дополнительные способы использования данных из базы данных. В результате работы с модулем вы научитесь:

- Форматировать записи базы данных на Web-странице
- Добавлять компонент, позволяющий загружать файлы в базу данных из Web-страницы
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Когда будете готовы, начните с модуля 1, Создание Web-страниц с соединениями с данными.

Необходимое время

Общее: 3 часа 30 минут

Модуль 1: 2 часа

Модуль 2: 1 час 30 минут

Отображение динамических сведений на Web-страницах с JavaServer Faces

Это учебник по разработке динамического Web-сайта, работающего как раздел рекламы и объявлений в газете. Web-сайт - это полное Web-приложение J2EE, использующее компоненты JavaServer Faces (JSF) и Service Data Objects (SDO). Web-приложение применяет технологию JSF для создания динамических Web-страниц, содержащих ссылки на базу данных с объявлениями.

Для работы с этим учебником может потребоваться установка дополнительных компонентов. Обратитесь к разделу системных требований и проверьте, все ли дополнительные компоненты установлены.

Это учебник по разработке простого динамического Web-сайта, моделирующего раздел рекламы и объявлений в газете. На сайте будут перечислены продаваемые предметы; пользователь сможет добавлять новые предметы, изменять сведения о предметах (например, цену или описание), а также искать предметы определенного типа. Хотя данный сайт весьма прост, рассматриваемые в нем принципы и технологии используются и в значительно более крупных и сложных Web-сайтах.

Для конструирования ссылок Web-страниц на базу данных, в которой хранятся объявления, в учебнике применяется технология JavaServer Faces (JSF). Переноса компоненты JSF на свои страницы, вы можете создавать Web-приложение JSF, называемое приложением CRUD (create-read-update-delete), поскольку оно может создавать, считывать, обновлять и удалять записи в базе данных. Пользователи, просматривающие Web-сайт, смогут применять эти функции для управления объявлениями в базе данных.

Цели обучения

Этот учебник разбит на два модуля, каждый из которых преследует свои цели обучения. Вы можете изучить любой из них или оба модуля. В каждом модуле вы должны выполнить упражнения, чтобы Web-сайт работал правильно.

Модуль 1. Создание Web-страниц с соединениями с данными

Благодаря работе с данным модулем вы узнаете, как настроить соединение с базой данных и использовать информацию из нее на Web-страницах. В результате работы с модулем вы научитесь:

- Подключать Web-страницы к базе данных.
- Создавать страницы, способные отображать, создавать, изменять и удалять записи из базы данных.
- Создавать базовую страницу результатов поиска.
- Отправлять данные с одной страницы на другую.

Модуль 2. Добавление дополнительных функций

В этом модуле описаны дополнительные способы использования данных из базы данных. В результате работы с модулем вы научитесь:

- Форматировать записи базы данных на Web-странице
- Добавлять компонент, позволяющий загружать файлы в базу данных из Web-страницы
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Когда будете готовы, начните с модуля 1, Создание Web-страниц с соединениями с данными.

Необходимое время

Общее: 3 часа 30 минут

Модуль 1: 2 часа

Модуль 2: 1 час 30 минут

Уровень пользователя

Промежуточный

Аудитория

Разработчики Web-приложений, проектировщики пользовательских интерфейсов Web

Требования к системе

Для работы с учебником необходимо предварительно установить и настроить сервер среды выполнения. Настоящий учебник работает со следующими серверами:

- WebSphere Application Server 6.0
- WebSphere Application Server 6.1
- WebSphere Application Server Community Edition
- Apache Tomcat 5.5

Для работы с этим учебником необходимо, чтобы в системе был установлен и настроен сервер приложений. Для того чтобы проверить, доступна ли среда выполнения сервера, щелкните на **Окно → Параметры**, разверните **Сервер** и щелкните на **Установленные среды выполнения**. С помощью показанной панели вы можете добавлять, удалять и изменять определения установленных сред выполнения сервера. Вы также можете загрузить и установить поддержку нового сервера.

Предварительные требования

Для работы с данным учебником необходимы:

- Знание основных концепций Web-дизайна - Web-сайтов, Web-страниц, браузеров и серверов.
- Умение создавать простые статические Web-страницы.
- Знакомство с элементами Web-страниц, такими как таблицы, гиперссылки, формы и изображения.
- Знакомство с основными терминами баз данных, такими как таблицы, записи, столбцы и поля.

Также нелишним будет понимание следующих принципов:

- Как применять проекции и панели в рабочей среде.
- Как редактировать код HTML Web-страниц.

Ожидаемые результаты

По окончании работы над Web-приложением домашняя страница будет выглядеть так, как показано на следующем рисунке. Ссылки на этой странице будут открывать страницы, предназначенные для создания новых объявлений, обновления или удаления существующих объявлений и поиска объявлений по категории.

Связанная информация

Просмотрите версию этого учебника в формате PDF.

Модуль 1. Создание Web-страниц с соединениями с данными

При работе с этим модулем вы научитесь создавать Web-страницы, позволяющие отображать, редактировать, удалять и создавать информацию из базы данных. Вы познакомитесь с технологией JavaServer Faces (JSF) и с ее помощью научитесь просто выполнять сложные операции. Задайте краткое описание для данного модуля учебника. Модуль учебника - это набор взаимосвязанных уроков.

Цели обучения

Благодаря работе с данным модулем вы узнаете, как настроить соединение с базой данных и использовать информацию из нее на Web-страницах. В результате работы с модулем вы научитесь:

- Подключать Web-страницы к базе данных.
- Создавать страницы, способные отображать, создавать, изменять и удалять записи из базы данных.

- Создавать базовую страницу результатов поиска.
- Отправлять данные с одной страницы на другую.

Необходимое время

Для прохождения данного модуля потребуется примерно 2 часа. Если во время работы с учебником вы захотите изучить другие аспекты динамических Web-сайтов, то это время может увеличиться.

Урок 1.1. Импортирование необходимых ресурсов и настройка целевого сервера

Перед началом работы необходимо импортировать необходимые ресурсы для этого учебника. Ресурсы включают набор Web-страниц и пример базы данных Derby.

Основная цель этого учебника - обучить вас созданию Web-приложений, предназначенных для работы с данными в базе данных. В учебнике не рассматриваются подробно особенности проектирования внешнего вида Web-сайта. Дизайн Web-сайта уже подготовлен для вас.

В этом упражнении вам предстоит импортировать подготовленные Web-страницы, CSS и шаблоны страниц. Кроме того, вы настроите целевой сервер для динамического Web-проекта.

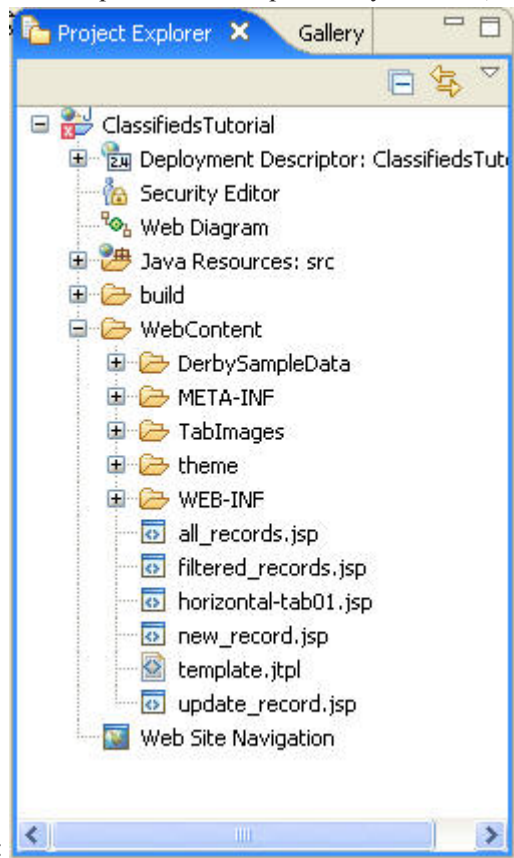
Для просмотра этого учебника вам понадобится доступ к существующему набору файлов JSP JavaServer Faces и пример базы данных Derby. Эти базовые Web-страницы и пример базы данных поставляются в виде ZIP-файла. В данном уроке вы познакомитесь с процессом импорта файла ZIP для применения Web-страниц и баз данных. Вы также узнаете, как просматривать импортированные файлы в рабочей среде.

Web-страницы и база данных, которые понадобятся вам при работе с данным учебником, поставляются в виде ZIP-файла. Для того чтобы импортировать содержимое ZIP-файла, выполните следующие действия:

Импортируйте пример файла проекта

1. Импортируйте проект ClassifiedsTutorial .

2. В панели Меню проектов Web-проекции убедитесь, что проект ClassifiedsTutorial выглядит следующим



образом:

Задайте целевой сервер для динамического Web-проекта

Указание целевого сервера для Web-проекта позволит тестировать ресурсы, которые вы будете создавать при работе с учебником.

Для того чтобы задать целевой сервер, выполните следующие действия:

1. В панели Меню проектов Web-проекции щелкните правой кнопкой мыши на **ClassifiedsTutorial** и выберите **Свойства**.
2. В списке свойств выберите **Сервер**.
3. В списке Сервер по умолчанию выберите сервер, который вы хотите применять по умолчанию. Нажмите **Применить**.
4. В списке свойств выберите **Целевые среды выполнения**.
5. В списке Среда выполнения щелкните на среде, соответствующей выбранному серверу. Нажмите **ОК**.

Если нужный сервер отсутствует в списке целевых сред выполнения, закройте рабочую среду и установите нужный сервер. После установки сервера выполните инструкции по указанию целевого сервера.

Примечание: Если список Сервер по умолчанию пуст, но вы установили среды выполнения сервера, то, возможно, требуется настроить сервер. Настроить сервер можно следующим образом:

1. Щелкните правой кнопкой мыши на файле **all_records.jsp**, затем выберите **Выполнить как** → **Выполнить на сервере**.
2. Выберите **Вручную определить новый сервер**.
3. Выберите установленный вами сервер.

Если вы теперь вернетесь к ранее описанной процедуре указания целевого сервера, то заметите, что только что настроенный вами сервер будет сервером по умолчанию. Если сервер установлен, но не настроен, то он не будет показан в списке серверов, в котором можно выбрать целевой сервер по умолчанию.

Контрольные точки урока

Вы импортировали динамический Web-проект `ClassifiedsTutorial` и задали целевой сервер.

Вы можете просмотреть файлы в Web-проекте учебника. Для того чтобы открыть файл, дважды щелкните на нем в панели Меню проектов. Для того чтобы просмотреть схему взаимосвязей страниц, дважды щелкните на Навигация по Web-сайту в Меню проектов.

Хотя в проекте теперь появилось много ресурсов - изображений, файлов кода и файлов конфигурации, в этом примере вы, в основном, будете работать со следующими файлами:

all records.jsp

Это домашняя страница сайта. На ней показаны все объявления из базы данных.

filtered_records.jsp

Это страница результатов поиска, отображающая только записи выбранной категории.

new_record.jsp

Эта страница создаст новое объявление.

update_record.jsp

Эта страница служит для изменения или удаления объявления из базы данных.

template.jtpl

Это шаблон страниц сайта. Он содержит такие элементы, как таблица и серый баннер "Добро пожаловать в службу объявлений", присутствующие на каждой странице. Под серым баннером на странице находятся две навигационные вкладки, позволяющие перейти к домашней странице и к странице добавления нового объявления.

Теперь вы готовы к тому, чтобы перейти к упражнению 1.2: Работа со списками реляционных записей и компонентами таблицы данных.

Урок 1.2. Подключение к базе данных и отображение данных на Web-странице

В этом уроке вы узнаете, как подключаться к базе данных и отображать текстовые записи на Web-странице. Вы также узнаете, как добавить список реляционных записей на Web-страницу.

На Web-сайте в данном учебнике для доступа к источникам данных, таким как базы данных, и отображения информации из этих источников на странице используются динамические Web-страницы с компонентами `JavaServer Faces`. В этом учебнике используются реляционные записи и списки реляционных записей для представления данных в базе данных, чтобы данные можно было просмотреть на странице в виде таблицы данных или обычной таблицы HTML. Этими компонентами используются объекты доступа Java.

Дополнительные сведения об объектах доступа к данным:

Объекты доступа к данным - это объекты Java, представляющие объекты EJB. Обычно они применяются в клиентских программах, работающих с файлами JSP, сервлетами или EJB, взаимодействующими с другими EJB. Объекты доступа скрывают от вас сложности управления жизненными циклами объектов EJB. Это означает, что вы можете программировать объекты EJB так же легко, как объекты Java, что упрощает клиентские программы объектов EJB и сокращает общее время разработки.

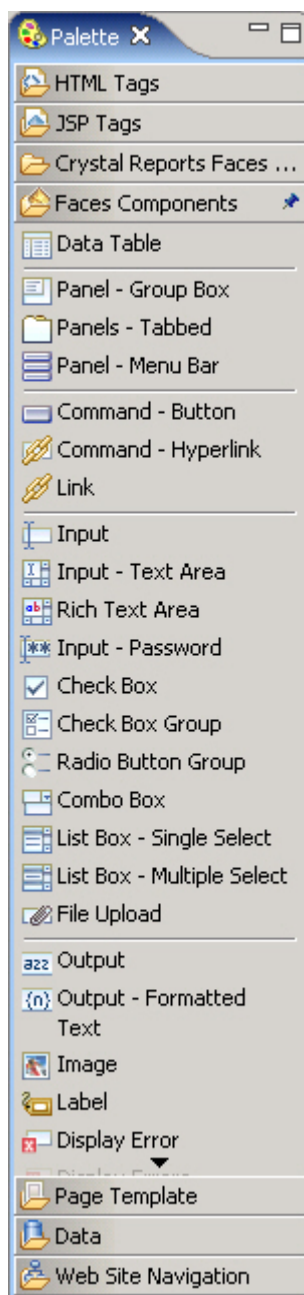
Дополнительные сведения о JavaServer Faces и компонентах Faces:

JavaServer Faces - это технология, разработанная фирмой Sun Microsystems. Она помогает создавать пользовательские интерфейсы для динамических Web-приложений, работающих на сервере. Среда JavaServer Faces управляет состоянием пользовательского интерфейса с помощью запросов сервера и предлагает простую модель для разработки событий сервера, активируемых клиентом. JavaServer Faces основан на среде модель-панель-контроллер (MVC). Применительно к JavaServer Faces это означает, что контроллер - это сервлет, модель представлена объектами JavaBeans, а панель состоит из компонентов JavaServer Faces с небольшими фрагментами кода приложений или вообще без него. Цель этой модели - отделить содержимое от представления.

Применять эту технологию в Web-приложениях помогают такие инструменты, как Компоненты Faces. Компоненты Faces содержат библиотеку пользовательских тегов JavaServer Page (JSP), что позволяет реализовывать интерфейс JavaServer Faces на странице. Этот мастер помогает вам создавать файлы JSP, поддерживающие использование компонентов Faces. Компоненты Faces позволяют разрабатывать Web-приложение путем переноса компонентов из лотка Faces в панель Палитра и размещения их на создаваемых Web-страницах.

Например, вы можете перенести текстовое поле ввода и поместить его на форму на странице. Затем вы можете перенести кнопку отправки и положить ее рядом с текстовым полем ввода. Наконец, вы можете соединить текстовое поле ввода с источником данных. Это позволит конечным пользователям вводить данные из Web-приложения в источник данных.

Другое преимущество компоновки приложений с помощью компонентов Faces заключается в том, что пользовательский интерфейс выводится независимо от базовой программной логики. Это означает, что приложения могут работать с данными на разных платформах, таких как браузеры или мобильные устройства.



Добавьте список реляционных записей

В этом уроке вы создадите список реляционных записей для представления всех объявлений в базе данных. Затем вы подключитесь к базе данных и выберите таблицу, содержащую сведения, необходимые для списка реляционных записей. Наконец, вы добавите список реляционных записей на страницу в таблице данных.

- Реляционные записи подключаются только к одной записи из базы данных. В таком случае реляционная запись представляет одно объявление из базы данных. С помощью реляционной записи можно создать одну запись, изменить существующую запись или удалить старую запись.
- Список реляционных записей подключается к нескольким записям из базы данных. В таком случае список реляционных записей представляет от одного до всех объявлений из базы данных.

- В таблицах данных отображаются данные из списка реляционных записей на странице. Таблицы данных просто являются местом хранения списков записей, они не форматируют данные в строки и столбцы, как это делает таблица HTML.

Дополнительные сведения о таблицах данных:

Таблица данных является компонентом на странице Faces JSP, содержащим объекты данных, такие как список реляционных записей. При добавлении списка реляционных записей мастер добавления списка реляционных записей добавляет таблицы данных автоматически. Можно также переместить таблицу данных из блока Расширенные компоненты Faces на Палитру, если требуется создать таблицу вручную.

Несмотря на то, что таблица с данными имеет строки и столбцы, подобно таблице HTML, она не является таблицей в полном смысле этого слова. При необходимости форматирования управляющих элементов ввода и вывода вам понадобится поле Панель - Группа из блока Компоненты Faces на Палитре. Этот вопрос описан в разделе Упражнение 2.2: Форматирование таблицы данных.

Так как таблицы данных являются компонентами Faces, а не компонентами HTML, они управляются с помощью панели Свойства, а не панели Проектировщик страниц. С помощью панели Свойства можно настроить таблицу данных различными способами:

- Добавить, удалить и изменить расположение столбцов
- Форматировать текст и фон
- Добавить заголовок, нижний колонтитул и поля
- Добавить страницы для отображения результатов

Для создания списка реляционных записей выполните следующие действия:

1. В панели Меню проектов разверните **Динамические Web-проекты** → **ClassifiedsTutorial** → **Web-содержимое**.
2. В папке Web-содержимое дважды щелкните на **all_records.jsp**. Файл all_records.jsp откроется в редакторе.
3. Удалите текст по умолчанию Добавьте содержимое.
4. В панели Палитра разверните **Блок данных**.
5. Переместите компонент **Список реляционных записей** из Палитры в пустую область содержимого. Откроется окно Добавить список реляционных записей.
6. В поле Имя введите all_recordlist. Имена списка реляционных записей и самих записей должны соответствовать стандарту имен переменных Java (например, они не могут содержать пробелы).
7. Убедитесь, что выбрана опция **Добавить управляющие элементы данных**. Если опция **Добавить управляющие элементы данных** выбрана, мастер создаст таблицу данных для отображения списка записей на странице. В ином случае мастер создаст только список записей без представления данных на странице. Сейчас же мастер создаст таблицу данных по умолчанию, которую вы сможете настроить в дальнейшем. Окно Добавить список реляционных записей выглядит примерно следующим образом:

Add Relational Record List

Relational Record List

Enter a reference name for record list.

Name:

Create a name to refer to this record list within the page.

☐ Retrieve an existing record or record list from scope

Scope:

Key:

☐ Reuse metadata definition from an existing record or record list

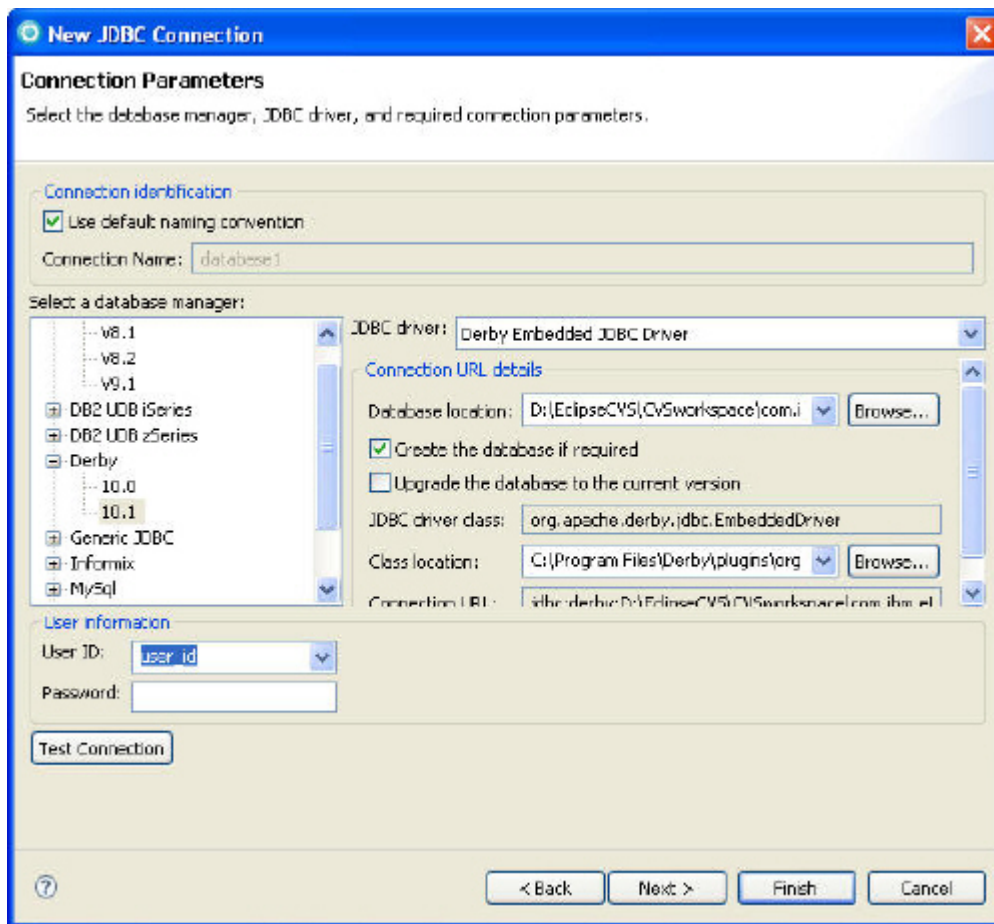
Input file:

You can automatically add data controls to your page to display the list of records (configure details on last page of this wizard).

☒ Add data controls

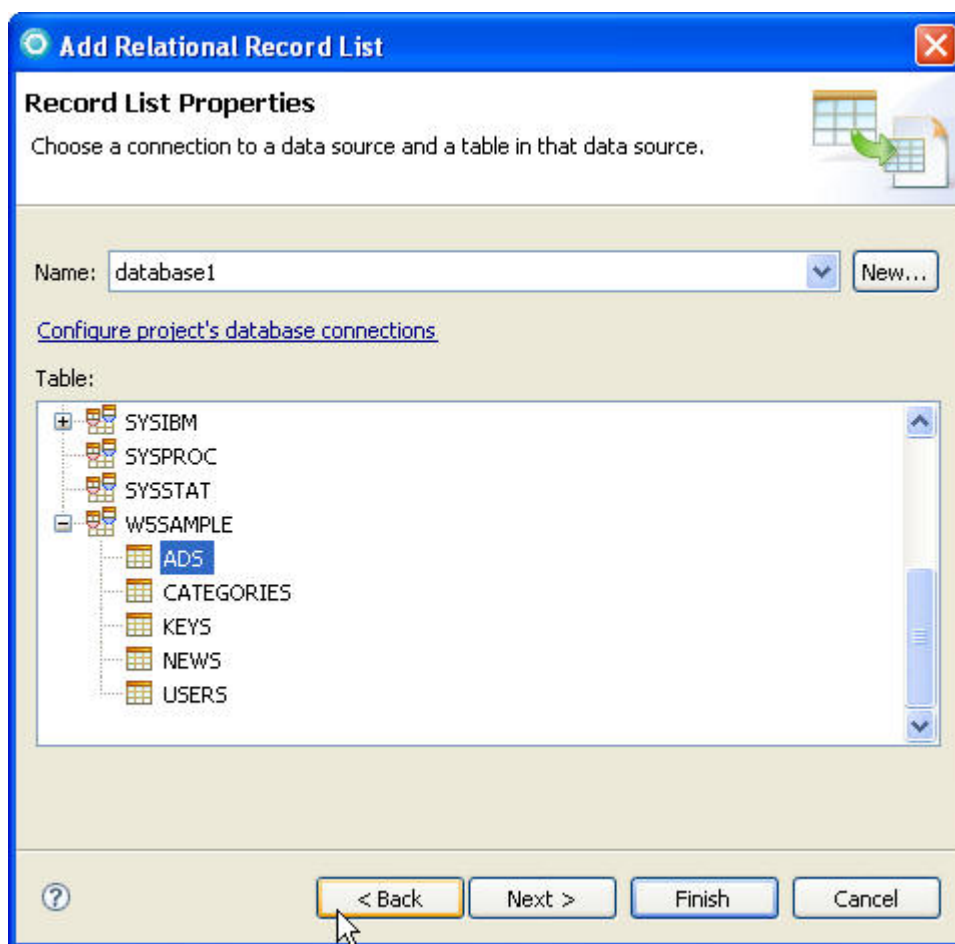
Нажмите кнопку **Далее**.

8. В поле **Имя соединения** на следующей странице мастера нажмите кнопку **Создать** для создания соединения с базой данных. Откроется окно Создать соединение JDBC. Выберите радиокнопку **Создать соединение** и нажмите **Далее**.
9. Нажмите **Выбрать администратор базы данных**, разверните **Derby** и выберите **10.1**.
10. В поле **Расположение базы данных** нажмите **Обзор** и выберите `<расположение_рабочей_области>\ClassifiedsTutorial\WebContent\DerbySampleData\database`, где `<расположение_рабочей_области>` - это каталог текущей рабочей области. Нажмите **ОК**, чтобы закрыть окно Обзор папок.
11. В поле **Расположение класса** введите расположение файла или файлов JAR, реализующих класс драйвера JDBC, или нажмите **Обзор**, чтобы выбрать расположение в файловой системе. Выбрав расположение файлов JAR, нажмите **Открыть**, чтобы закрыть окно Открыть.
12. Для доступа к базе данных может потребоваться ввести ИД пользователя. Пароль не нужен. Мастер Создать соединение с базой данных будет выглядеть следующим образом:



13. В окне мастера Создать соединение нажмите **Готово**. Вы вернетесь в окно мастера Добавить список реляционных записей. Теперь, создав соединение с базой данных Derby, необходимо выбрать таблицу или список записей для представления. Мастер Добавить список реляционных записей покажет таблицы из базы данных.

Разверните **W5SAMPLE** и выберите **ADS**. Нажмите кнопку **Далее**. На оставшихся страницах мастера можно добавить столбцы для списка записей и настроить дополнительные опции, например задать главный ключ, взаимосвязи с другими таблицами и указать фильтры и критерии сортировки. Более подробно об этих страницах вы сможете узнать при выполнении других упражнений.



14. На странице Добавить реляционную запись окна вы укажете столбцы, которые вы хотите включить в Web-страницу. Столбец Идентификатор будет выбран автоматически, и удалить его отметку нельзя; однако вы сможете сделать это на следующем шаге. При работе с данным учебником вам понадобится отобразить лишь заголовок, описание, категорию, цену и телефонный номер для каждого объявления. По умолчанию выбраны все столбцы; оставьте только следующие столбцы:

- **НАЗВАНИЕ:** String
- **ОПИСАНИЕ:** String
- **ГЛАВНАЯ КАТЕГОРИЯ:** String
- **ЦЕНА:** BigDecimal
- **ТЕЛЕФОН:** String

Нажмите кнопку **Далее**.

15. На странице Настроить управляющие элементы данных окна вы можете задать дополнительные опции, уточняющие отображение таблицы данных на Web-странице. Для того чтобы переупорядочить столбцы, выберите столбец и с помощью кнопок со стрелками поместите его в нужное место.

Щелкните на столбце **ИД**, чтобы удалить его из таблицы.

Расположите столбцы в следующем порядке:

- **НАЗВАНИЕ**
- **ОПИСАНИЕ**
- **ГЛАВНАЯ КАТЕГОРИЯ**
- **ЦЕНА**
- **ТЕЛЕФОН**

Нажмите кнопку **Готово**. Web-страница будет выглядеть следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#)

[Post New Listing](#)

Maincategory ^{abc}	Title ^{abc}	Description ^{abc}	Price ^{abc}	Phone ^{abc}
{MAINCATEGORY} ^{abc}	{TITLE} ^{abc}	{DESCRIPTION} ^{abc}	{PRICE} ¹²³	{PHONE} ^{abc}

16. Сохраните страницу.

В следующем уроке вы узнаете, как будет выглядеть эта страница после опубликования на Web-сервере. Для форматирования таблиц с данными и компонентов JavaServer Faces существует несколько вариантов. Некоторые из них будут описаны в следующем модуле. Можно также просмотреть панель Свойства для различных компонентов JavaServer Faces (например, таблицы с данными и отдельные компоненты ввода).

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы узнали, как подключаться к базе данных и отображать записи данных на Web-странице с помощью списка реляционных записей.

Урок 1.3. Тестирование Web-сайта

Когда ваше Web-приложение будет готово к опубликованию, вам потребуется сервер для хостинга, чтобы пользователи смогли обращаться к Web-сайту через Internet, однако для тестирования сайта можно воспользоваться средой тестирования для имитации сервера.

В данном уроке вы выполните JSP в среде тестирования для проверки внешнего вида страницы после ее опубликования. Также вы узнаете, как запускать и останавливать сервер. В любое время в ходе разработки Web-сайта вы сможете открыть страницу в Проектировщике страниц и на вкладке **Предварительный просмотр** посмотреть, как она будет выглядеть в браузере. Однако панель Предварительный просмотр не позволяет проверить динамические аспекты страницы (например, соединение с базой данных) и узнать, как они будут выполняться на сервере.

Совет: Если сервер запущен, то перед продолжением работы с сайтом необходимо его остановить.

Для работы с этим учебником в системе должен быть установлен и настроен сервер приложений. Для того чтобы проверить, доступна ли среда выполнения сервера, щелкните на **Окно → Параметры**, разверните **Сервер** и щелкните на **Установленные среды выполнения**. С помощью показанной панели вы можете добавлять, удалять и изменять определения установленных сред выполнения сервера. Вы также можете загрузить и установить поддержку нового сервера.

Для проверки Web-сайта на сервере выполните следующие действия:

1. В панели Меню проектов щелкните правой кнопкой мыши на файле **all_records.jsp**, затем выберите **Выполнить как → Выполнить на сервере**. Откроется окно Выполнить на сервере
2. Необходимо использовать тот же сервер, что был выбран в качестве целевого в уроке 1.1. Применение другого сервера вызовет ошибки.
 - Для проверки Web-сайта в ранее определенной конфигурации сервера выполните следующие действия:
 - a. Выберите **Выбрать существующий сервер**.
 - b. Выберите сервер из списка.
 - Для проверки Web-сайта в новой конфигурации сервера выполните следующие действия:
 - a. Выберите **Вручную определить новый сервер**.
 - b. Выберите тип сервера из списка.
3. Нажмите **Готово**.

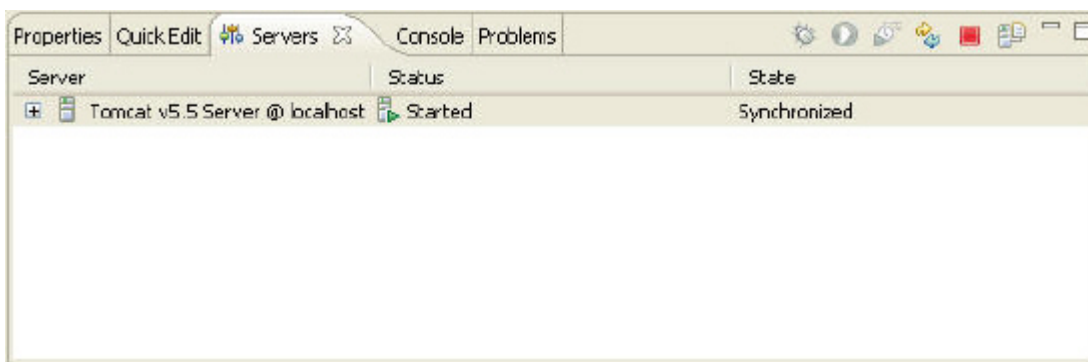
Инструменты сервера создадут новый сервер, запустят его и откроют страницу во внутреннем Web-браузере рабочей среды. Это может занять пару минут. В ходе запуска сервера в панели Консоль будут появляться сообщения.

Теперь вы можете заранее проверить, как будет выглядеть и работать сайт после его опубликования на Web-сервере. Можно щелкать на ссылках, вводить данные в формы и просматривать динамически создаваемое содержимое сайта, недоступное в панели Предварительный просмотр. Если необходимо проверить страницу в различных Web-браузерах, то можно скопировать URL из Web-браузера рабочей среды в любой другой браузер.

Перед продолжением выполнения упражнений необходимо остановить сервер. Если оставить сервер работающим, при продолжении выполнения упражнений возникнут ошибки. Эти ошибки возникнут из-за того, что сервер подключен к примеру базы данных, что мешает подключиться к базе данных для изменения информации, отображаемой на Web-сайте.

Для остановки сервера выполните следующие действия:

1. Перейдите на вкладку Серверы. Эта панель обычно расположена внизу страницы по центру. Если вы не видите панель Серверы, выберите **Окно → Показать панель → Серверы** в главном меню.
2. Выберите сервер, который требуется остановить. Заметьте, что состояние сервера указано как *Запущен*.



3. Нажмите кнопку **Остановить сервер**,  расположенную в правом верхнем углу панели Серверы. **Состояние** сервера изменится на *Останавливается*. В панели Консоль появятся сообщения, отражающие процесс завершения работы сервера. Когда состояние сервера в панели Серверы изменится на **Остановлен**, сервер будет остановлен и вы сможете продолжить работу с Web-сайтом.

Совет: Вы можете проверить Web-сайт на сервере, но не забудьте после этого остановить сервер.

Контрольные точки урока

Вы закончили выполнение этого урока. В данном уроке вы проверили JSP в среде тестирования, чтобы выяснить, как будет выглядеть страница после ее опубликования. Вы также научились запускать и останавливать сервер.

Урок 1.4. Создание записи для отображения и обновления базы данных

В этом уроке вы создадите страницу, позволяющую пользователям создавать новые объявления и помещать их в базу данных.

Вначале создайте реляционную запись, представляющую новую запись в базе данных, затем создайте визуализацию записи на странице в виде набора полей ввода.

Для создания списка реляционных записей выполните следующие действия:

1. Дважды щелкните на файле **new_record.jsp** в панели Меню проектов, чтобы открыть его.
2. Удалите текст по умолчанию **Добавьте содержимое**.
3. В панели Палитра щелкните на блоке **Данные**, чтобы развернуть его.
4. Переместите компонент **Реляционная запись** из Палитры в пустую область содержимого. Откроется окно **Добавить реляционную запись**.

Примечание: Появление предупреждения с сообщением о невозможности установить соединение с базой данных означает, что вы оставили сервер работающим после тестирования Web-сайта. В таком случае нажмите кнопку **Отмена** во всех окнах и остановите сервер, как описано в Уроке 1.3. .

5. В поле **Имя** введите `create_record`.
6. В пункте **Создать управляющие элементы для** выберите **Создание записи**.
7. Убедитесь, что опция **Добавить управляющие элементы ввода-вывода для отображения записи на Web-странице** выбрана. Окно **Добавить реляционную запись** будет выглядеть следующим образом:

8. Нажмите **Далее**.
9. В поле Таблица разверните **W5SAMPLE.ADS** и выберите ADS. Нажмите **Далее**.
10. Нажмите **Далее** еще раз. Откроется страница Настроить управляющие элементы данных.

Совет: На странице Настроить управляющие элементы данных можно настроить визуализацию реляционных записей. Например, можно изменить столбцы, метки полей и кнопку отправки формы ввода. Выполнив эти действия, вы получите полнофункциональную страницу ввода.

11. В области **Отображаемые поля** снимите выделение с переключателями рядом с именем поля, за исключением тех, которые должны присутствовать в форме ввода:
 - ОПИСАНИЕ
 - ИД
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ТЕЛЕФОН
 - ЦЕНА
 - НАЗВАНИЕ

12. Измените порядок размещения полей с помощью клавиш Вверх  и Вниз 
 - ИД

- НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
- Измените метки. Например, можно сократить метку "Главная категория:" до "Категория:". Для того чтобы изменить метки, созданные для полей ввода, щелкните на столбце **Метка**. Значок указателя мыши превратится в курсор, и вы сможете ввести текст.
 - Выберите **ГЛАВНАЯ КАТЕГОРИЯ** и щелкните на **Опции**. Откроется окно Опции.
 - Убедитесь, что опция **Кнопка отправки** выбрана.
 - Введите Добавить новый список в поле **Метка**. Затем нажмите **ОК**. Окно Добавить реляционную запись будет выглядеть следующим образом:

Add Relational Record

Configure Data Controls

Specify the columns to display and how to display them

Fields to display:

Field Name	Label	Control Type
☒ ID (java.lang.Integer)	Id:	Input Field
☒ TITLE (java.lang.String)	Title:	Input Field
☒ DESCRIPTION (java.lang.String)	Description:	Input Field
☒ MAINCATEGORY (java.lang.String)	Category:	Input Field
☒ PRICE (java.math.BigDecimal)	Price:	Input Field
☒ PHONE (java.lang.String)	Phone:	Input Field
☐ TYPECODE (java.lang.Short)	Typecode:	Input Field
☐ DATEOPEN (java.util.Date)	Dateopen:	Input Field
☐ CATEGORY_ID (java.lang.Integer)	Category_id:	Input Field
☐ TYPE (java.lang.String)	Type:	Input Field
☐ EMAIL (java.lang.String)	Email:	Input Field
☐ PHOTO (byte[])	Photo:	File Upload
☐ IMAGETYPE (java.lang.String)	Imagetype:	Input Field
☐ LOCATION (java.lang.String)	Location:	Input Field

All None Options...

? < Back Next > Finish Cancel

- Нажмите **Готово** для создания формы ввода. Она будет выглядеть следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#)[Post New Listing](#)

Id:	{ID} 123
Title:	{TITLE} abc
Description:	{DESCRIPTION} abc
Category:	{MAINCATEGORY} abc
Price:	{PRICE} 123
Phone:	{PHONE} abc

{Error Messages} ↓

Post New Listing

Примечание: В форме есть поле Сообщения об ошибках. Это не значит, что в проекте есть ошибки; это поле помечает место, где будут показаны сведения об ошибках, если они возникнут, при отправке формы пользователем.

18. Сохраните страницу.

При отправке формы ввода страница автоматически добавит новую запись в базу данных. В следующем уроке вы запрограммируете кнопку **Добавить новый список**. Эта кнопка позволяет вернуться к странице `all_records.jsp` и сразу же просмотреть новую запись в базе данных.

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы создали страницу, позволяющую пользователям создавать новые объявления и помещать их в базу данных.

Урок 1.5. Программирование кнопки отправки

При отправке формы ввода страница автоматически добавит новую запись в базу данных. Вы можете запрограммировать кнопку **Добавить новый список**, которая позволяет вернуться к странице `all_records.jsp` и сразу же просмотреть новую запись в базе данных.

В предыдущем уроке вы создали реляционную запись, позволяющую обновлять базу данных. В этом уроке вы запрограммируете кнопку отправки **Добавить новый список**, позволяющую просматривать обновления в базе данных.

Для того чтобы запрограммировать кнопку отправки, выполните следующие действия:

1. В файле `new_record.jsp` щелкните правой кнопкой мыши на кнопке **Добавить новый список**, созданной в форме на Web-странице, и выберите **Свойства**.

2. Нажмите **Добавить правило**. Откроется окно Добавить правило навигации.
3. В выпадающем списке Страница выберите **all_records.jsp**, затем нажмите **ОК**.
4. Сохраните страницу. Если вы хотите попробовать добавить запись в базу данных, чтобы убедиться, что вы вернетесь к странице all_records.jsp, то запустите new_record.jsp на сервере тестирования.

Следите за тем, чтобы ключи не совпадали

Важная информация:

Поскольку столбец ИД - это главный ключ в таблице ADS, вы не можете добавлять записи со значением ИД, уже существующим в таблице. В Модуле 2 вы увидите, как с помощью функции автоматического создания ключей можно автоматически создавать уникальный ключ для каждой новой записи.

Однако пока вы должны ввести уникальный ИД на этой странице, чтобы добавить новую запись. Записи, поставляемые вместе с базой данных, содержат ИД с номерами от 1 до 22, так что вы можете ввести любое число больше 22 в качестве главного ключа. Следите за тем, чтобы ключи не совпадали, если вы вводите несколько записей.

Привяжите ввод к реляционной записи (необязательно)

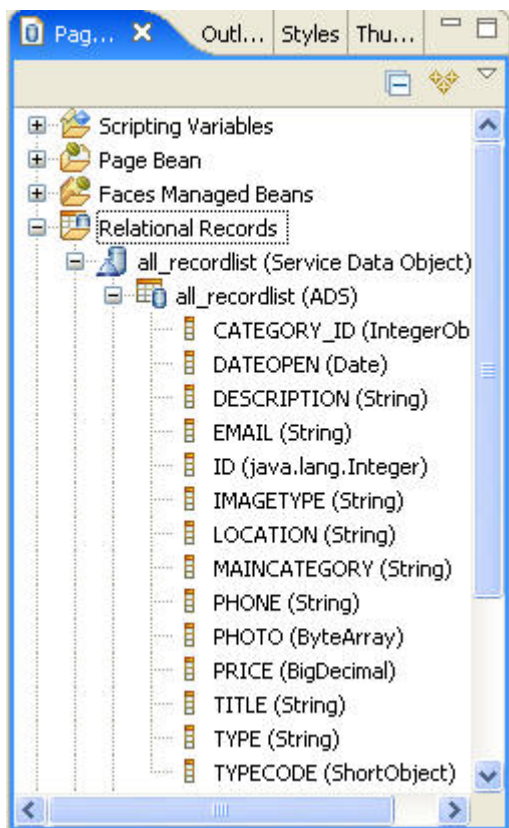
Форма ввода на странице - это набор полей ввода JavaServer Faces, связанных с созданной вами реляционной записью. Вспомните, что вы создали реляционную запись для представления новой записи в таблице ADS базы данных. Привязывание - это метод, позволяющий привязать компонент ввода JavaServer Faces к столбцу в реляционной записи.

Когда вы создали реляционную запись, мастер автоматически связал все ее столбцы с полями ввода на странице. Если вы хотите внести изменения вручную, то вы можете связать остальные столбцы с другими полями ввода. Для того чтобы связать столбец из реляционной записи с полем ввода, перенесите столбец из панели Данные страницы в поле. Вы можете поэкспериментировать с этим процессом, удаляя и вновь создавая поле ввода Описание в своей форме.

Дополнительные сведения о панели Данные страницы:

Панель Данные страницы обычно находится в левом нижнем углу рабочей области. Если вы не видите ее, перейдите в строку меню и выберите **Окно → Показать панель → Данные страницы**. Обычно панель Данные страницы используется только при создании динамических Web-страниц.

Панель Данные страницы хранит соединения с источниками данных, такими как базы данных, в форме объектов данных, таких как реляционные записи или списки реляционных записей. Эти объекты данных не представляют ни сам источник данных, ни какой-либо компонент на странице. Эти объекты данных представляют соединение между проектом и источником данных. После создания соединения вы можете перенести объект данных из панели Данные страницы на Web-страницу, чтобы закончить создание соединения и отобразить информацию из источника данных на Web-странице. Например, на следующем рисунке изображена панель Данные страницы со списком реляционных записей, созданным во время работы с этим учебником:



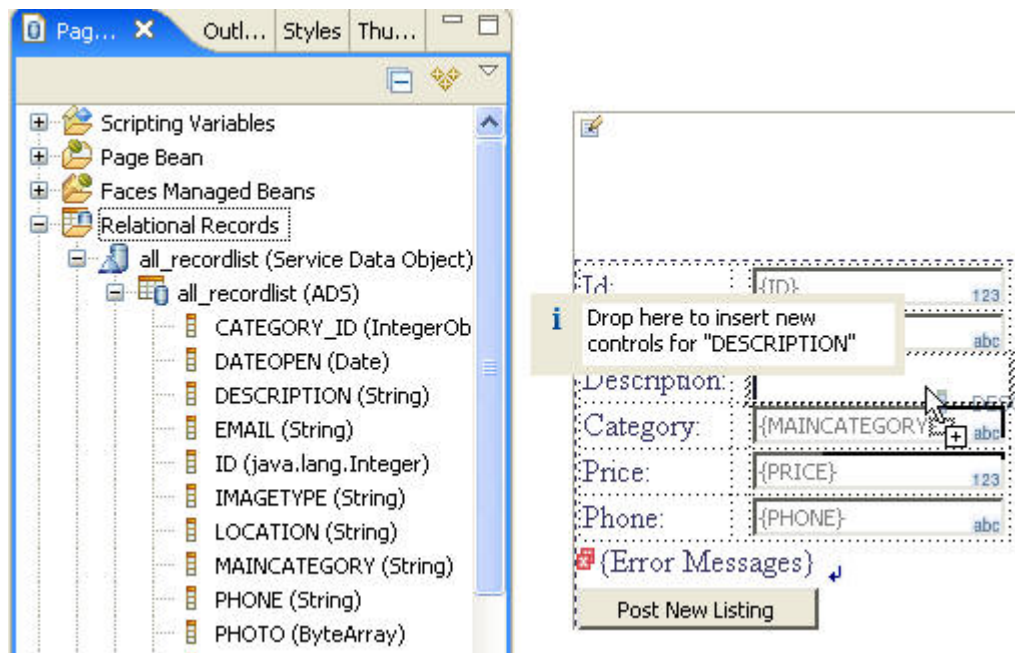
На этом рисунке изображен список реляционных записей `all_recordlist (ADS)`, за которым следуют все столбцы этого списка. После того как этот объект данных определен и появился в панели Данные страницы, вы можете переносить эти столбцы на компоненты ввода на Web-странице столько раз, сколько необходимо, не подключаясь заново к базе данных.

Например, если вы хотите отобразить значение столбца ОПИСАНИЕ на Web-странице, то сначала перенесите текстовое поле вывода на страницу, а затем перенесите столбец ОПИСАНИЕ из панели Данные страницы на это текстовое поле вывода. Это называется связыванием столбца с полем вывода.

Или, если вы хотели позволить пользователю ввести новое значение в столбце ОПИСАНИЕ в базе данных, вы можете перенести текстовое поле ввода на страницу. Затем перенесите столбец ОПИСАНИЕ на это текстовое поле ввода.

Следующие действия иллюстрируют концепцию поля ввода и процесс связывания; однако выполнять эту процедуру необязательно при работе с учебником. Если вы не хотите выполнять эти действия, перейдите к следующему уроку.

1. В файле `new_record.jsp` выберите поле ввода **Описание**. Нажмите клавишу **Delete**.
2. В панели Палитра щелкните на блоке **Расширенные компоненты Faces**, чтобы развернуть его.
3. Перенесите компонент **Ввод** из Палитры в ячейку, в которой находилось только что удаленное поле ввода **Описание**. Теперь в этой ячейке есть поле ввода, но нет текста, такого как {ИД} или {ЗАГОЛОВОК}, поскольку это поле ввода не связано ни с одним столбцом.
4. В панели Данные страницы разверните **Реляционные записи** → **create_record (Service Data Object)** → **create_record (ADS)** и перенесите столбец ОПИСАНИЕ на только что созданный компонент **Ввод**. Текст внутри компонента **Ввод** изменится и теперь будет указывать, что он связан со столбцом ОПИСАНИЕ, как показано ниже:



5. Сохраните файл и затем запустите страницу на сервере тестирования, если хотите.

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы научились программировать командную кнопку и связывать данные с полем ввода путем переноса виджета JSF из Палитры на Web-страницу.

Урок 1.6. Создание страницы обновления

На данный момент, вы создали страницы для просмотра и создания списков для Web-сайта. В этом упражнении вы создадите страницу, позволяющую пользователям обновлять и удалять списки. Страница обновления будет выглядеть практически точно так же, как страница создания, за исключением того, что на странице обновления поля ввода будут содержать данные из существующей записи, которые пользователь может изменить.

Сначала вы создадите реляционную запись, представляющую существующую запись из базы данных. Затем вы создадите форму обновления JavaServer Faces для этой реляционной записи, и после внесения нескольких небольших исправлений ваша страница будет готова к тестированию.

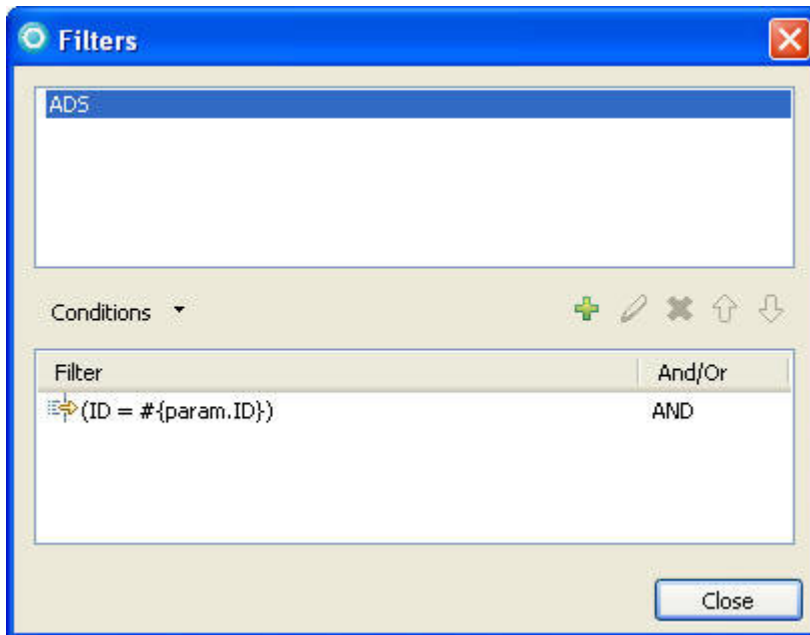
Создать реляционную запись обновления

1. Откройте файл **update_record.jsp**, дважды щелкнув на нем в панели Меню проектов.
2. Удалите текст по умолчанию **Добавьте содержимое**.
3. В панели Палитра щелкните на блоке **Данные**, чтобы развернуть его.
4. Переместите компонент **Реляционная запись** из Палитры в пустую область содержимого. Откроется окно **Добавить реляционную запись**.
5. В поле **Имя** введите **update_record**.
6. В поле **Создать управляющие элементы для** выберите **Обновление существующей записи**.
7. Убедитесь, что опция **Добавить управляющие элементы ввода-вывода для отображения записи на Web-странице** выбрана.
8. Нажмите **Далее**.
9. В поле **Таблица** разверните **W5SAMPLE** и выберите **ADS**.
10. Нажмите кнопку **Далее**. Откроется страница **Добавить реляционную запись**.

Отфильтруйте результаты

Реляционная запись может показывать только одну запись из базы данных. Таким образом, вы должны отфильтровать таблицу базы данных, чтобы пользователю для редактирования была показана только одна запись. (Вам не требовалось фильтровать базу данных в предыдущем упражнении, поскольку вы создали новую запись, а следовательно, в базе данных не было результатов, которые можно было бы отфильтровать.) Поскольку каждой записи в базе данных присвоен уникальный числовой ИД, вы отфильтруете результаты, оставив один результат с заданным номером.

1. На правой стороне страницы Добавить реляционную запись, в разделе **Задачи**, выберите **Фильтровать результаты**. Откроется окно Фильтры с условием фильтра по умолчанию ID = #{param.ID} в столбце **Фильтр**. Окно Фильтры выглядит так:



Этот код фильтрует записи в базе данных, так что в реляционной записи будет присутствовать только запись с указанным числовым ИД. Дополнительную информацию об этом условии вы получите в разделе Вставка гиперссылки позже в этом упражнении. Нажмите **Заккрыть**.

2. Нажмите кнопку **Далее**. Откроется страница Настроить управляющие элементы данных.
3. В области **Отображаемые поля** удалите отметки с переключателей всех полей, кроме тех, которые должны присутствовать в форме ввода:
 - ИД
 - НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
4. С помощью клавиш **Вверх** и **Вниз** переупорядочьте имена полей сверху вниз в следующем порядке:
 - ИД
 - НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН

5. Для поля ИД выберите **Поле вывода** в выпадающем списке в столбце **Тип управляющего элемента**. Пользователям следует разрешить просматривать числовой ИД записи, но запретить обновлять его. Вставка поля ИД в поле Вывод поможет вам избежать проблемы неуникальности идентификаторов.
6. Нажмите **Опции**. Откроется окно Опции. Убедитесь, что опция **Кнопка отправки** выбрана. Введите Обновить в поле **Метка**. Нажмите **ОК**. Окно Добавить реляционную запись выглядит примерно следующим образом:

Add Relational Record List

Configure Data Controls

Specify the columns to display and how to display them

Data control to create: Multi-Column Data Table (one table row per data entry)

Columns to display:

Column Name	Label	Control Type
☒ TITLE (java.lang.String)	Title	Output Field
☒ DESCRIPTION (java.lang.String)	Description	Output Field
☒ MAINCATEGORY (java.lang.String)	Maincategory	Output Field
☒ PRICE (java.math.BigDecimal)	Price	Output Field
☒ PHONE (java.lang.String)	Phone	Output Field
☐ ID (java.lang.Integer)	Id	Output Field

All None Options...

< Back Next > Finish Cancel

7. Нажмите **Готово**, чтобы создать форму обновления на странице, как показано ниже:

Welcome to the Classifieds!

[View All Classifieds](#)[Post A Listing](#)

Id:	{ID} 123
Title:	{TITLE} abc
Description:	{DESCRIPTION} abc
Maincategory:	{MAINCATEGORY} abc
Price:	{PRICE} 123
Phone:	{PHONE} abc

{Error Messages} ↓

Update Delete

Запрограммируйте кнопку обновления

На этот раз вы добавите код на страницу `all_records.jsp`, чтобы измененная запись отображалась вместе со всеми остальными записями.

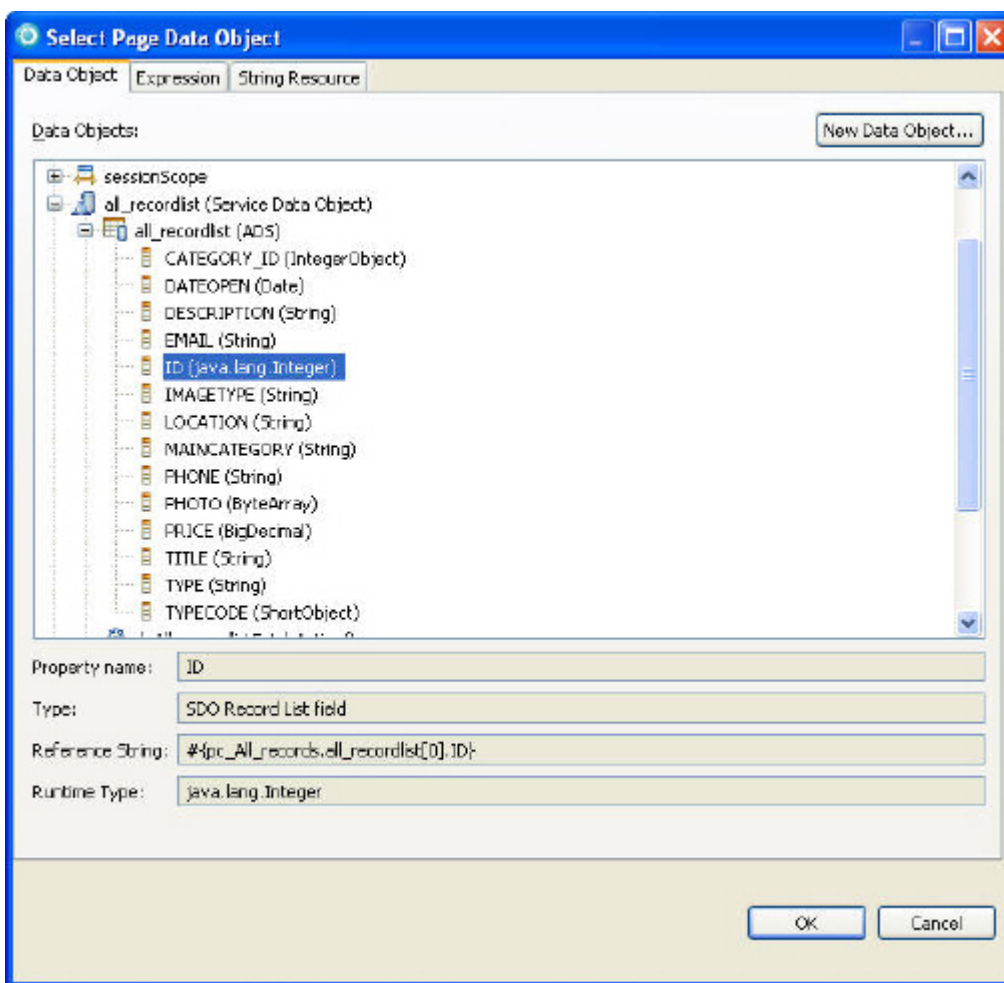
1. Щелкните правой кнопкой мыши на кнопке **Обновить**, которую вы только что создали на Web-странице, и выберите **Свойства**.
2. Нажмите **Добавить правило**. Откроется окно Добавить правило навигации.
3. В выпадающем списке Страница выберите **all_records.jsp**, затем нажмите **OK**.
4. Повторите шаги 1-3, чтобы так же отредактировать код для кнопки Delete на Web-странице.
5. Сохраните страницу. Если вы хотите попробовать добавить запись в базу данных, чтобы убедиться, что вы вернетесь к странице `all_records.jsp`, то запустите `new_record.jsp` на сервере тестирования.

Вставьте гиперссылку

Теперь вы создадите ссылки на странице `all_records.jsp`, так что пользователь сможет выбрать запись базы данных для обновления. `# {param.ID}` представляет числовой ИД записи, которую обновит страница обновления. Когда пользователь выбирает ссылку на запись, этот числовой ИД записи отправляется на страницу `update_record.jsp` в качестве параметра `# {param.ID}`. Затем отфильтрованная реляционная запись, которую вы только что вставили на страницу `update_record.jsp`, появится как единственная доступная для обновления запись.

1. В панели Меню проектов дважды щелкните на файле **all_records.jsp**, чтобы открыть его в редакторе. Щелкните в любом месте таблицы данных. Откройте панель Свойства.
2. В панели Свойства выберите **hx:dataTableEx** в списке тегов HTML в левой части панели.
3. Нажмите **Добавить** для добавления нового столбца. Введите Обновить в поле Метка нового столбца. В таблице данных будет создан столбец Обновить.

4. Столбец Обновить должен стоять последним в строке. Для переупорядочения столбцов в таблице данных можно воспользоваться кнопками **Переместить вверх** и **Переместить вниз** в панели `hx:dataTableEx`.
5. В панели Палитра разверните блок **Расширенные компоненты Faces**.
6. Перенесите компонент **Ссылка** из Палитры в только что созданный столбец Обновить. Появится окно Настроить URL. Нажмите **ОК**, чтобы закрыть окно.
7. Щелкните на гиперссылке в столбце Обновить.
8. В панели Свойства выберите вкладку **Параметр**, расположенную сразу под тегом `hx:outputLinkEx` в левой части панели.
9. Нажмите **Добавить параметр** и затем введите ИД в поле Имя. Вы должны связать параметр ИД со столбцом ИД в форме ввода. Связывание параметра гиперссылки со столбцом ИД списка данных означает, что при выборе гиперссылки параметром запроса будет ИД записи в списке.
10. Выберите поле **Значение** и нажмите кнопку **Выбрать объект данных страницы**  в поле. Откроется окно Выбрать объект данных страницы.
11. На вкладке **Объекты данных** выберите **all_recordlist (Service Data Object)** → **all_recordlist (ADS)** → **ИД (java.lang.Integer)**, как показано ниже:



Нажмите **ОК**. Теперь, когда пользователь щелкнет на ссылке Обновить список, Web-сайт позволит пользователю обновить информацию об объявлении.

12. Сохраните файл и проверьте страницу, если хотите. Помните, что сначала нужно открыть `all_records.jsp`, поскольку это страница, связанная с `update_record.jsp`.

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы узнали, как создавать Web-страницы, обновляющие записи в базе данных.

Урок 1.7. Фильтрация списка реляционных записей

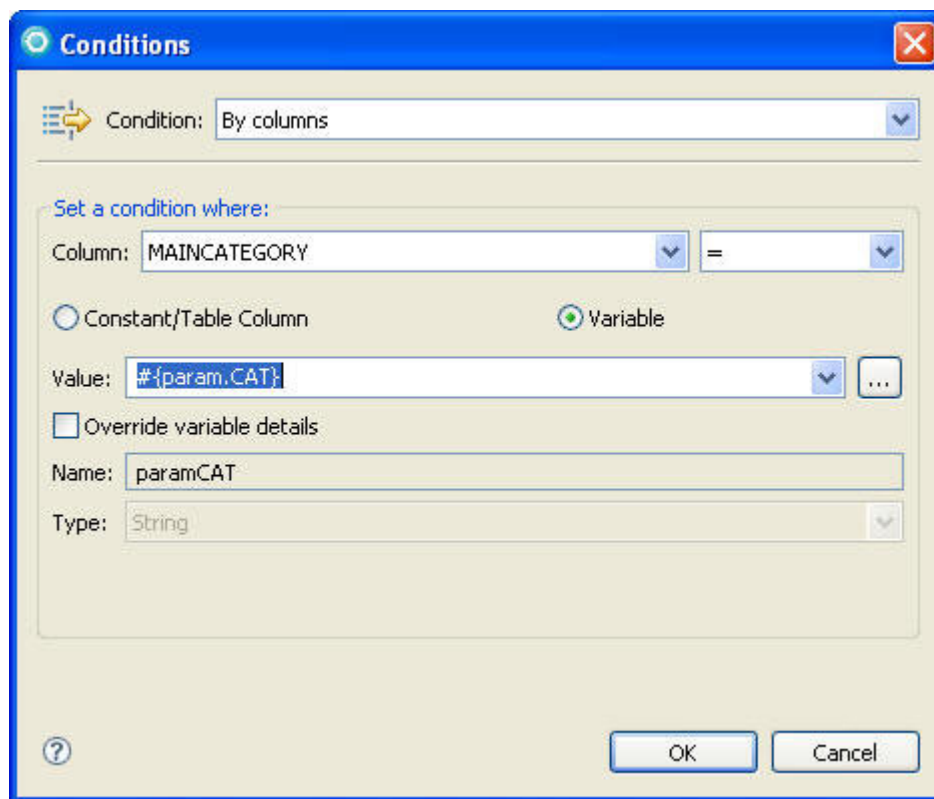
Вы уже знаете, как использовать список реляционных записей для отображения всех записей в таблице и как применять условие фильтра в реляционной записи для выделения отдельной записи. Теперь вы научитесь отображать только те записи, которые отвечают конкретному критерию; благодаря этому пользователи смогут просматривать объявления по категориям.

Как в случае страницы обновления, вы будете фильтровать по параметру, содержащему требуемую категорию, передаваемую на страницу по гиперссылке. Поскольку категории может соответствовать несколько записей, фильтр будет применяться к списку реляционных записей, а не к одной реляционной записи.

Создайте отфильтрованный список реляционных записей

Создание отфильтрованного списка реляционных записей включает дополнительный шаг фильтрации списка, в результате которого остаются лишь нужные записи. Как и при создании страницы `update_record.jsp`, вы укажете параметр на странице `all_records.jsp` и с помощью ссылки отправите этот параметр на страницу `filtered_records.jsp`.

1. Откройте файл **filtered_records.jsp**, дважды щелкнув на нем в панели Меню проектов. Удалите текст по умолчанию.
2. Создайте список реляционных записей, перенеся **Список реляционных записей** из блока Данные Палитры на страницу. Откроется окно Добавить список реляционных записей. Назовите этот список записей `filtered_recordlist`. Убедитесь, что выбрана опция **Добавить управляющие элементы данных**, и нажмите **Далее**.
3. Разверните **W5SAMPLE** и выберите **ADS**. Нажмите **Далее**.
4. В разделе Задачи нажмите **Результаты фильтрации**. Откроется окно Фильтры. Для создания условия фильтра нажмите кнопку **Добавить**  на вкладке **Условия**. Откроется окно Условия. Выберите столбец **ГЛАВНАЯ КАТЕГОРИЯ** в выпадающем списке **Столбец**. Отметьте радиокнопку **Переменная**.
5. Введите `#{param.CAT}` в поле **Значение**. Во время выполнения этот код будет вычислен и станет значением параметра запроса CAT. Окно Условия должно выглядеть следующим образом:



Нажмите **ОК**, чтобы закрыть окно Условия, затем **Заккрыть**, чтобы закрыть окно Фильтры. Нажмите **Далее**.

6. Вновь у вас есть выбор между различными столбцами, которые могут быть показаны в таблице данных. Выберите только следующие столбцы:
 - НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
7. Расположите столбцы в следующем порядке:
 - НАЗВАНИЕ
 - ОПИСАНИЕ
 - ГЛАВНАЯ КАТЕГОРИЯ
 - ЦЕНА
 - ТЕЛЕФОН
8. Окно Добавить список реляционных записей выглядит примерно следующим образом:

Add Relational Record List

Configure Data Controls

Specify the columns to display and how to display them

Data control to create: Multi-Column Data Table (one table row per data entry)

Columns to display:

Column Name	Label	Control Type	
☒ TITLE (java.lang.String)	Title	Output Field	
☒ DESCRIPTION (java.lang.String)	Description	Output Field	
☒ MAINCATEGORY (java.lang.String)	Maincategory	Output Field	
☒ PRICE (java.math.BigDecimal)	Price	Output Field	
☒ PHONE (java.lang.String)	Phone	Output Field	
☐ ID (java.lang.Integer)	Id	Output Field	

Нажмите **Готово**. Список реляционных записей будет добавлен на страницу. Он выглядит точно так же, как список записей на странице `all_records.jsp`, но, поскольку он отфильтрован по категории, он будет показывать записи только одной категории в каждый момент времени.

- Сохраните страницу. Страница будет выглядеть следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#)

[Post New Listing](#)

Current photo:

Id: (ID) 123

Title: {TITLE} abc

Description: {DESCRIPTION} abc

Maincategory: {MAINCATEGORY} abc

Price: {PRICE} 123

Phone: {PHONE} abc

(Error Messages) ↓

Update Delete

Ссылка из all_records.jsp

Как и в случае страницы обновления, теперь вы должны создать гиперссылку в файле all_records.jsp, которая будет не только открывать файл filtered_records.jsp, но и передавать параметр CAT для фильтрации списка реляционных записей.

1. Откройте файл **all_records.jsp**. В панели Палитра разверните блок **Расширенные компоненты Faces**.
2. Перенесите компонент **Ссылка** из Палитры непосредственно на управляющий элемент вывода {ГЛАВНАЯ КАТЕГОРИЯ} в таблице данных. Откроется окно Настроить URL.
3. В поле **URL** введите `filtered_records.jsp` в качестве целевой страницы и оставьте поле **Метка** пустым. Если ссылка перенесена непосредственно на управляющий элемент вывода, а поле **Метка** оставлено пустым, то гиперссылка будет использовать текст {ГЛАВНАЯ КАТЕГОРИЯ} в качестве своей метки. Нажмите **ОК**.
4. Щелкните на ссылке, а затем щелкните на вкладке **Параметр** в разделе **hx:outputLinkEx** панели Свойства.
5. Нажмите **Добавить параметр** для создания параметра. Введите **CAT** в поле **Имя для нового параметра**.
6. Для того чтобы связать параметр **CAT** со столбцом **ГЛАВНАЯ КАТЕГОРИЯ** в форме ввода, выберите поле **Значение** и затем нажмите кнопку **Выбрать объект данных страницы** в этом поле. Откроется окно **Выбрать объект данных страницы**.
7. На вкладке **Объекты данных** разверните **all_recordlist (Service Data Object) → all_recordlist (ADS) → ГЛАВНАЯ КАТЕГОРИЯ (String)**. Нажмите **ОК**.
8. Сохраните страницу и проверьте ее. При щелчке на имени категории вы перейдете на страницу `filtered_records.jsp`, на которой показаны только объявления, относящиеся к этой категории.

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы научились отображать только те записи, которые отвечают конкретному критерию; благодаря этому пользователи смогут просматривать объявления по категориям.

Модуль 1 - итоги

Вы закончили работу с модулем 1, Создание Web-страниц с соединениями с данными.

Пройденные уроки

Теперь вы умеете:

- Отображать информацию из баз данных на Web-страницах.
- Работать с реляционными записями, списками реляционных записей и таблицами данных.
- Отображать, редактировать, создавать и удалять записи базы данных из Web-страницы.
- Создавать ссылку на запись или набор записей посредством гиперссылки и компонентов поля со списком.

Этот модуль послужил введением в технологию и инструменты JavaServer Faces. Существует множество других операций, позволяющих сделать ваш сайт привлекательным, удобным и эффективным. В следующем модуле, “Модуль 2. Добавление дополнительных функций” на стр. 31, вы научитесь превращать этот простой сайт объявлений в более визуально привлекательное и сложное приложение. Перейдите к следующему модулю, если вы хотите научиться проектировать сайт, так чтобы он выглядел следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#)[Post a Listing](#)

TITLE	DESCRIPTION	CATEGORY
Video Camera 	Description: Great camera, suitable for amateur or professional use. Price: \$900.00 Phone: 534-333-454 Update Listing	Entertainment
VCR 	Description: Old and worn but still working ok, hence low price. Price: \$40.00 Phone: 322-654-044 Update Listing	Entertainment
Flat Screen TV 	Description: Very clear picture, sound excellent too. Reluctant sale. Price: \$700.00 Phone: 456-456-944 Update Listing	Entertainment
Mini Stereo 	Description: Small stereo with a big sound. Price: \$100.00 Phone: 543-453-345 Update Listing	Entertainment
72 Les Paul Custom 	Description: Excellent condition, first to see will buy. Price: \$3,000.00 Phone: 345-333-543 Update Listing	Entertainment

[1](#) [2](#) [3](#) [4](#) [5](#)

Модуль 2. Добавление дополнительных функций

В этом модуле вы научитесь пользоваться дополнительными функциями для изменения формата Web-страниц, добавлять функцию, позволяющую пользователям загружать файлы в базу данных, настраивать правила, автоматически направляющие пользователей к нужной странице, и автоматизировать выполнение некоторых административных задач, например генерации ключей.

Цели обучения

В этом модуле описаны дополнительные способы использования данных из базы данных. В результате работы с модулем вы научитесь:

- Форматировать записи базы данных на Web-странице
- Добавлять компонент, позволяющий загружать файлы в базу данных из Web-страницы
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Необходимое время

Для прохождения данного модуля потребуется примерно полтора часа. Если во время работы с учебником вы захотите изучить другие аспекты динамических Web-сайтов, то это время может увеличиться.

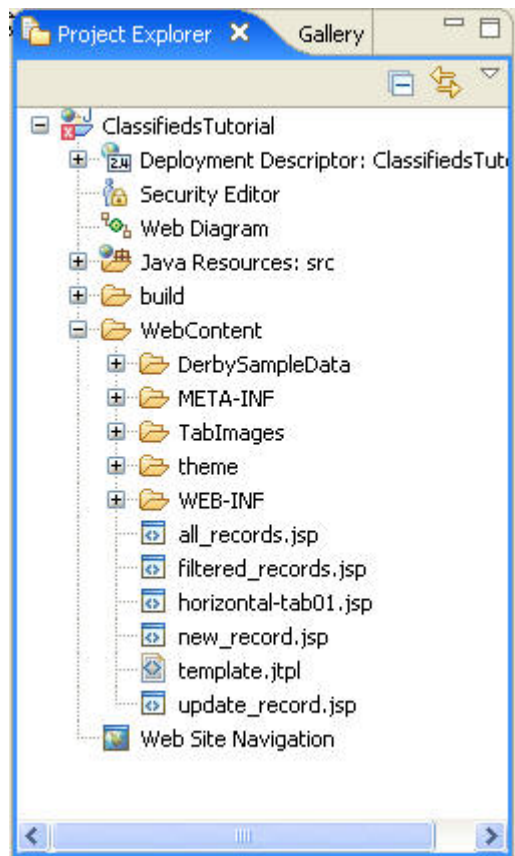
Предварительные требования

Если вы уже закончили работу с модулем 1, Создание Web-страниц с соединениями с данными, то пропустите оставшуюся часть этого раздела предварительных требований и начните работу с разделом “Урок 2.1. Форматирование таблицы данных” на стр. 34.

Если вы начинаете работу с этим учебником с модуля 2, пропустив модуль 1, то сначала вы должны импортировать необходимые ресурсы, настроить целевой сервер и настроить соединение с базой данных.

Для того чтобы импортировать необходимые ресурсы, выполните следующие действия:

1. Импортируйте проект ClassifiedsTutorial .
2. В панели Меню проектов Web-проекции убедитесь, что проект ClassifiedsTutorial выглядит следующим образом:



Для того чтобы задать целевой сервер, выполните следующие действия:

1. В панели Меню проектов Web-проекции щелкните правой кнопкой мыши на **ClassifiedsTutorial** и выберите **Свойства**.
2. В списке свойств выберите **Сервер**.
3. В списке Сервер по умолчанию выберите сервер, который вы хотите применять по умолчанию. Нажмите кнопку **Применить**.
4. В списке свойств выберите **Целевые среды выполнения**.
5. В списке Среда выполнения щелкните на среде, соответствующей выбранному серверу. Нажмите кнопку **ОК**.

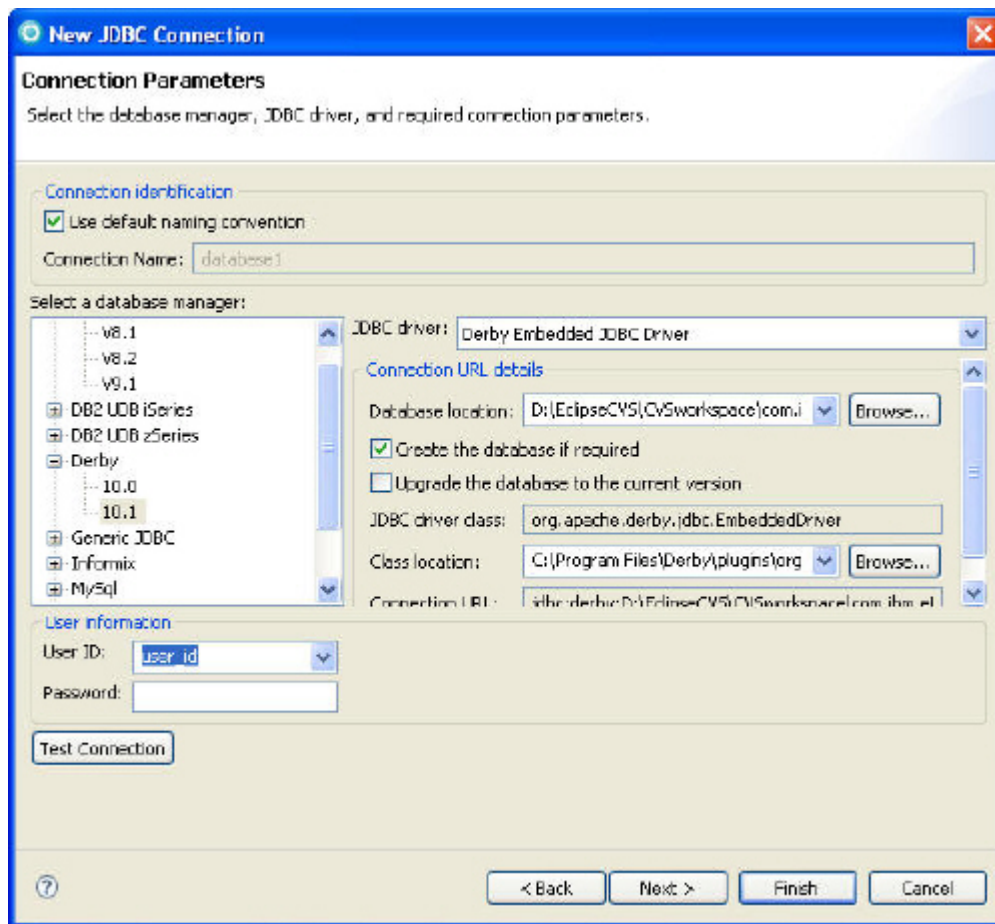
Если нужный сервер отсутствует в списке целевых сред выполнения, закройте рабочую среду и установите нужный сервер. После установки сервера выполните инструкции по указанию целевого сервера.

Для того чтобы настроить соединение с базой данных, выполните следующие действия:

1. В панели Меню проектов разверните **Динамические Web-проекты** → **ClassifiedsTutorial** → **Web-содержимое**.
2. В папке Web-содержимое дважды щелкните на **all_records.jsp**. В редакторе откроется файл all_records.jsp.
3. В панели Данные страницы дважды щелкните на **all_recordlist (Service Data Object)**. Появится окно Настроить список реляционных записей. Панель Данные страницы обычно находится в левом нижнем углу рабочей области. Если вы не видите панель Данные страницы, выберите **Окно** → **Показать панель** → **Данные страницы**, и она появится.
4. В окне Настроить список реляционных записей щелкните на вкладке **Соединения**.
5. Выберите **Создать**. Откроется окно Создать соединение.
6. Выберите **Создать соединение с базой данных**. Откроется окно Создать соединение с базой данных. Нажмите кнопку **Далее**.

7. Нажмите кнопку **Обзор**, расположенную рядом с полем Расположение базы данных. Откроется окно Выбор папки.
8. В разделе Выбрать каталог базы данных перейдите к следующему каталогу: <папка_рабочей_области>\ClassifiedsTutorial\WebContent\cloudscapesampleddata\database, где <папка_рабочей_области> - это каталог, содержащий текущую рабочую область.
9. Выберите папку **database** и нажмите **ОК**.

Для доступа к базе данных добавлять ИД пользователя или пароль не требуется. Окно Создать соединение с базой данных будет выглядеть следующим образом:



Вы можете просмотреть файлы в Web-проекте учебника. Для того чтобы открыть файл, дважды щелкните на нем в панели Меню проектов. Для того чтобы просмотреть схему взаимосвязей страниц, дважды щелкните на Навигация по Web-сайту в Меню проектов.

Хотя в проекте теперь появилось много ресурсов - изображений, файлов кода и файлов конфигурации, в этом примере вы, в основном, будете работать со следующими файлами:

all records.jsp

Это домашняя страница сайта. На ней показаны все объявления из базы данных.

filtered_records.jsp

Это страница результатов поиска, отображающая только записи выбранной категории.

new_record.jsp

Эта страница создаст новое объявление.

update_record.jsp

Эта страница служит для изменения или удаления объявления из базы данных.

template.jtpl

Это шаблон страниц сайта. Он содержит такие элементы, как таблица и серый баннер "Добро пожаловать в службу объявлений", присутствующие на каждой странице. Под серым баннером на странице находятся две навигационные вкладки, позволяющие перейти к домашней странице и к странице добавления нового объявления.

Урок 2.1. Форматирование таблицы данных

Таблица данных на странице `all_records.jsp` выглядит утилитарно. Она содержит все записи базы данных и ссылки на другие страницы, в чем и заключается ее предназначение, но она не выглядит привлекательно. В этом упражнении вы улучшите внешний вид этой таблицы, добавив в нее разбиение на страницы, стили и изображения продаваемых элементов.

Переименуйте столбцы

Зачастую названия столбцов в таблице с данными не должны совпадать с именами столбцов базы данных. В следующих шагах описывается, как переименовать столбцы таблицы данных.

1. Дважды щелкните на странице **all_records.jsp** в панели Меню проектов
2. Щелкните на заголовке столбца **Главная категория** в таблице данных.
3. В панели Свойства щелкните на вкладке **h:outputText** и измените текстовое поле **Значение** с Главная категория на Категория.
4. При желании переименуйте и другие столбцы.
5. Сохраните страницу

Форматирование компонентов вывода

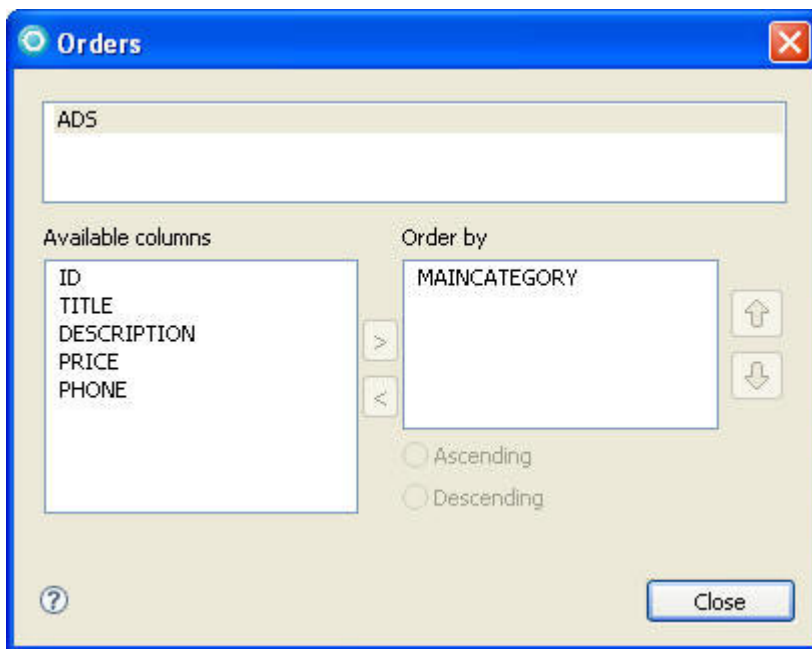
Вы можете отформатировать компоненты вывода. Далее мы отформатируем компонент вывода `{ЦЕНА}`, чтобы он отображался в виде денежной единицы, а не обычного числа.

1. Щелкните на компоненте вывода `{ЦЕНА}` на странице.
2. В панели Свойства найдите выпадающий список **Тип** и выберите Денежная единица. Теперь цена всех элементов будет отображена как денежная единица, а не обычное число. Для прочих типов данных доступны и другие варианты - проценты, дата и время. Можно также настроить формат компонента вывода для отображения типа данных, не перечисленных в панели Свойства, например, телефонный номер.
3. Сохраните страницу.

Упорядочьте данные

Сейчас данные показаны неупорядоченно. Для того чтобы создать более организованный список объявлений, упорядочьте его по категории, сгруппировав схожие элементы.

1. Щелкните правой кнопкой мыши на **all_recordlist (ADS)** в панели Данные страницы и выберите **Настроить** в контекстном меню. Откроется окно Настроить список реляционных записей. Появление предупреждения с сообщением о невозможности установить соединение с базой данных означает, что вы оставили сервер работающим после тестирования Web-сайта. В таком случае нажмите кнопку Отмена во всех окнах и остановите сервер, как описано в разделе "Остановка сервера" в упражнении Тестирование Web-сайта.
2. Отметьте первые два переключателя, "Извлечь существующую запись или список записей из области действия" и "Использовать повторно определение метаданных из существующей записи или списка записей", и нажмите **Далее**.
3. В следующем окне выберите ADS и нажмите **Далее**.
4. В правой части следующего окна выберите **Упорядочить результаты**. Откроется окно Упорядочение.
5. В панели Доступные столбцы выберите ГЛАВНАЯ КАТЕГОРИЯ.
6. Нажмите кнопку **>** для перемещения столбца ГЛАВНАЯ КАТЕГОРИЯ в область Упорядочить по панели. Окно должно выглядеть следующим образом:



7. Нажмите **Заккрыть**, чтобы закрыть окно Упорядочение.
8. Нажмите **Готово** для применения изменений. Теперь данные будут рассортированы по категориям. Сортировку в базе данных можно выполнить по любому столбцу.

Добавить разбиение на страницы

Вместо отображения каждой записи по одной в таблице с данными можно использовать разбиение на страницы. При этом записи автоматически разбиваются на страницы заданного размера без создания новых файлов JSP в проекте.

1. Щелкните в любом месте таблицы данных.
2. В панели Свойства выберите **h:dataTableEx** из списка тегов HTML в левой части панели.
3. Щелкните на вкладке **Опции отображения** в разделе **h:dataTableEx**.
4. В поле **Число строк на странице** введите 5.
5. Нажмите кнопку **Добавить разбиение на страницы**. Внизу таблицы с данными появится пример внешнего вида разбиения на страницы. Визуализация разбиения на страницы в Проектировщике страниц является заменителем и не отражает фактический вид, так как он будет виден только при загрузке страницы в браузере.
6. Сохраните страницу. Вы можете поэкспериментировать с разными типами разбиения на страницы и компонента статистики страниц для выбора наиболее подходящего варианта для страницы.

Разместите компоненты группы элементов в виде списка

Вместо ровно одного компонента вывода в каждом столбце таблицы с данными можно сочетать элементы со столбцами для получения более привлекательной разметки. С помощью групп элементов можно упорядочить компоненты точно так же, как и скрытые таблицы HTML. Например, добавив изображения для улучшения облика сайта, вы сможете поместить большую часть текстовой информации из объявлений в один столбец и назвать его СВЕДЕНИЯ.

1. Щелкните на блоке Расширенные компоненты Faces в Палитре, чтобы открыть его. Перенесите компонент **Панель - Группа элементов** в столбец НАЗВАНИЕ таблицы данных. Откроется окно **Выбрать тип**.
2. Выберите **Список** в качестве типа группы элементов в окне **Выбрать тип** и нажмите **ОК**. Эта группа элементов размещает компоненты, находящиеся внутри нее, в едином столбце или строке, и ее можно развернуть горизонтально или вертикально.
3. Щелкните на группе элементов, чтобы выбрать ее.

- С помощью панели Свойства измените **Ориентацию** на **Вертикальную**.
- Переместите компонент **{НАЗВАНИЕ}** в группу элементов "список". После добавления компонента в группу элементов поясняющий текст исчезнет.
- Перенесите компонент **Изображение** из блока Расширенные компоненты Faces Палитры в группу элементов "список". Рекомендуется поместить компонент изображения в конец группы, чтобы изображение было размещено под названием.
- Щелкните на только что добавленном компоненте изображения.
- С помощью области **Размер** панели Свойства задайте **Ширину** равной 60 пикселям, а **Высоту** равной 50 пикселям. Это обеспечит вывод изображений на странице размером 60x50 пикселей, вне зависимости от их истинного размера.
- Привяжите компонент изображения к столбцу ФОТОГРАФИЯ all_recordlist, перенеся столбец ФОТОГРАФИЯ из панели Данные страницы на компонент изображения. Это приведет к тому, что компонент изображения покажет данные изображения, обнаруженные в столбце ФОТОГРАФИЯ для каждой записи. Если столбец ФОТОГРАФИЯ (ByteArray) отсутствует, щелкните правой кнопкой мыши на **all_recordlist (ADS)** в панели Данные страницы и выберите **Настроить** в меню. Нажмите **ОК**, чтобы закрыть окно URL, если оно откроется. В окне Настроить список реляционных записей нажмите **Далее**. Выберите ADS и затем нажмите **Далее**. Отметьте переключатель **ФОТОГРАФИЯ (ByteArray)** и нажмите **Готово**.
- В окне Данные страницы разверните **all_recordlist (Service Data Object) → all_recordlist (ADS)**. Если **ТИП ИЗОБРАЖЕНИЯ (String)** присутствует, перейдите к шагу 12. Если столбец **ТИП ИЗОБРАЖЕНИЯ (String)** отсутствует, щелкните правой кнопкой мыши на **all_recordlist (ADS)** и выберите **Настроить** в меню.
- Нажмите **ОК**, чтобы закрыть окно URL, если оно откроется. В окне Настроить список реляционных записей нажмите **Далее**. Выберите ADS и затем нажмите **Далее**. Отметьте переключатель **ТИП ИЗОБРАЖЕНИЯ (String)** и нажмите **Готово**.
- Щелкните на вставленном вами изображении. На вкладке **hx:graphicImageEx** панели Свойства нажмите кнопку **Выбрать объект данных страницы**  рядом с полем **Тип**. Откроется окно Выбрать объект данных страницы.
- Разверните **all_recordlist (Service Data Object) → all_recordlist (ADS)**.
- Выберите **ТИП ИЗОБРАЖЕНИЯ (String)**.
- Нажмите **ОК**, затем сохраните страницу.

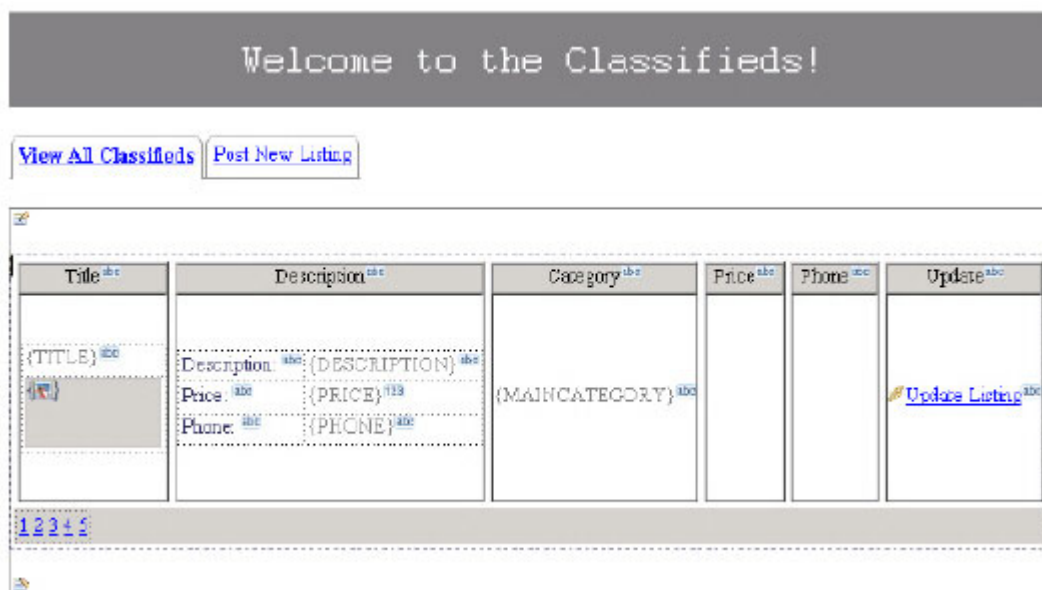
Ваша страница будет выглядеть следующим образом:



Уточните разметку с помощью сетки

Группа элементов позволяет организовать компоненты наподобие таблицы. Группа элементов типа "список" может содержать одну строку или столбец, но группа элементов типа "сетка" может содержать любое число строк и столбцов. В описанных ниже шагах мы переместим компоненты {ЦЕНА} и {ТЕЛЕФОН} в столбец ОПИСАНИЕ таблицы данных и добавим метки для них.

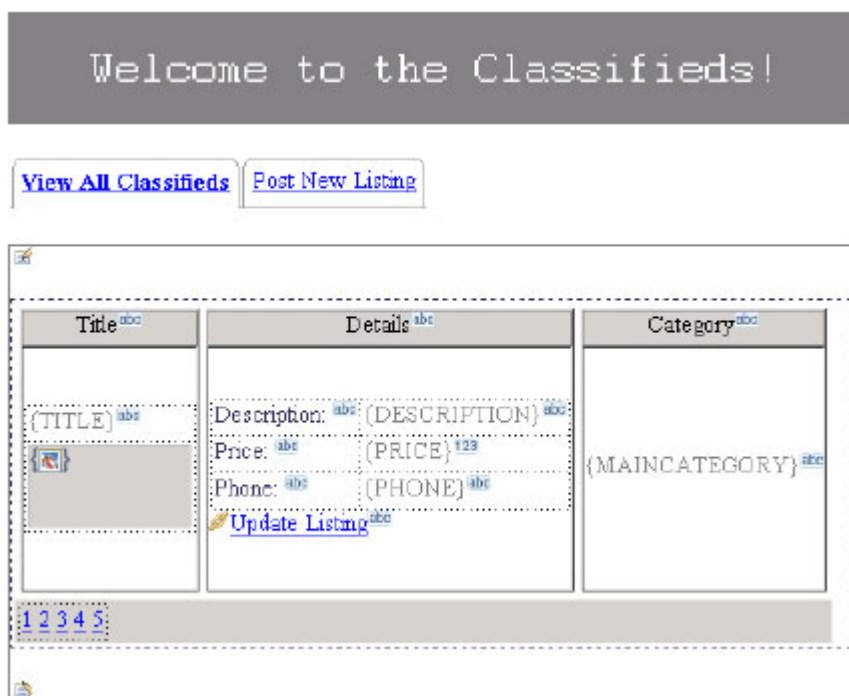
1. Перенесите компонент **Панель - Группа элементов** из блока Расширенные компоненты Faces Палитры в столбец ОПИСАНИЕ таблицы данных. Откроется окно **Выбрать тип**.
2. Выберите **Сетка** в качестве добавляемого типа компонента и нажмите **ОК**.
3. Щелкните на новой группе элементов "сетка".
4. В панели Свойства задайте число **Столбцов** для этой группы равным 2.
5. Перенесите компонент **Вывод** из блока Компоненты Faces Палитры на группу элементов "сетка". Этот компонент вывода станет меткой для описания продаваемого элемента. У каждого компонента вывода будет метка вроде этой, расположенная в левой ячейке таблицы.
6. Выберите компонент **Вывод** (он показан в Проектировщике страниц как **outputText**) и с помощью панели Свойства присвойте ему **Значение**, Описание:
7. Перенесите существующий компонент {ОПИСАНИЕ} к нижнему краю группы элементов "сетка". Если при перемещении компонентов в группе элементов у вас возникли затруднения, попробуйте нажать кнопку мыши и посмотреть на расположение курсора. Оно указывает местоположение компонента после перемещения. В данном упражнении необходимо отпустить кнопку мыши, когда курсор будет находиться справа от предыдущего компонента.
8. Перенесите другой компонент **Вывод** из Палитры и поместите его справа от группы элементов.
9. В панели Свойства присвойте ему **Значение** Цена:.
10. Поместите существующий компонент вывода {ЦЕНА} вправо от текста вывода Цена.
11. Таким же образом перенесите новый компонент **Вывод** для метки для компонента {ТЕЛЕФОН} и пометьте его как Телефон:.
12. Перенесите существующий компонент {ТЕЛЕФОН} в группу элементов "сетка" справа от текста вывода Телефон. Ваша страница будет выглядеть следующим образом:



13. Перенесите существующую гиперссылку **Обновить список** в столбец ОПИСАНИЕ, но *не* помещайте ее внутрь группы элементов "сетка". Лучше всего сделать это так: поместите указатель мыши на значок ссылки, затем, нажав и не отпуская кнопку мыши, перенесите ссылку непосредственно под группу элементов "сетка". Таким образом текстовая метка и значок ссылки будут перемещены, что не нарушит

работоспособность ссылки. Если переместить только ссылку или только текст, ссылка перестанет работать. Если это произойдет, выберите в меню пункт **Редактировать** → **Отменить** и повторите попытку.

14. Удалите пустые столбцы ЦЕНА, ТЕЛЕФОН и ОБНОВИТЬ из таблицы данных. Для удаления столбца щелкните на нем и откройте панель Свойства. Затем выберите **h:dataTable** из списка тегов HTML в левой части панели, выберите столбец, который вы хотите удалить, из списка в правой части панели и нажмите кнопку Удалить.
15. Щелкните на заголовке в столбце Описание и с помощью панели Свойства измените его поле **Значение** на Сведения. Страница будет выглядеть следующим образом:

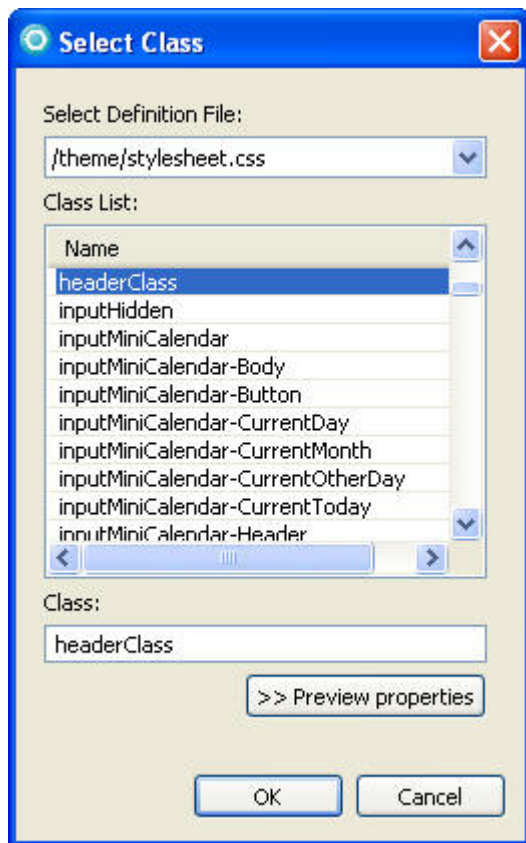


16. Сохраните страницу.

Примените стили

CSS не только является одним из лучших способов управления внешним видом Web-сайта, но также позволяет контролировать внешний вид отдельных компонентов JavaServer Faces. В следующем разделе вы воспользуетесь файлом CSS, поставляемым вместе с проектом, для изменения внешнего вида этой страницы. В частности, вы примените правило к заголовкам столбца и два правила к строкам для создания эффекта смены цветов, облегчающего восприятие строк таблицы.

1. Выберите таблицу данных, щелкнув на одном из заголовков столбцов.
2. В панели Свойства выберите **h:dataTableEx**.
3. Нажмите кнопку **Все атрибуты** , расположенную в правом верхнем углу панели Свойства.
4. В разделе **Имя атрибута** щелкните на элементе **headerClass**. Этот параметр отвечает за внешний вид строки таблицы.
5. Затем нажмите кнопку **Выбрать класс** , расположенную в поле Значение headerClass. Откроется окно Выбрать класс со всеми доступными стилями таблицы стилей проекта. Сейчас в таблице стилей доступен только один стиль.
6. Разверните таблицу стилей и щелкните на **.columnHeaderClass**.
7. Нажмите **ОК**.



8. Затем щелкните на элементе **rowClasses** под **Именем атрибута** и нажмите кнопку **Выбрать классы**  в поле Значение, чтобы вновь открыть окно Выбрать классы. Опция rowClasses отвечает за вид таблицы данных.
9. Разверните таблицу стилей и выберите оба правила **.rowClass1** и **.rowClass2**, удерживая нажатой клавишу **Ctrl** при щелчках на правилах. Эти два правила задают различные цветовые схемы, и выбор их обоих приведет к изменению цвета строк.
10. Нажмите **OK**.
11. Наконец, сохраните страницу и запустите ее на сервере тестирования. Если вы не умеете запускать приложения на сервере, обратитесь к разделу “Урок 1.3. Тестирование Web-сайта” на стр. 12. При

тестировании страницы она должна выглядеть следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#) [Post New Listing](#)

Title	Details:	Category
 Video Camera	Description: Great camera, suitable for amateur or professional use. Price: \$900.00 Phone: 534-333-454 Update Listing	Entertainment
 VCR	Description: Old and worn but still working ok, hence low price. Price: \$40.00 Phone: 322-654-044 Update Listing	Entertainment
 Flat Screen TV	Description: Very clear picture, sound excellent too. Refurbished only. Price: \$700.00 Phone: 456-456-944 Update Listing	Entertainment
 Mini Stereo	Description: Small stereo with a big sound. Price: \$100.00 Phone: 543-453-345 Update Listing	Entertainment
 72 Les Paul Custom	Description: Excellent condition, first to see will buy. Price: \$3,000.00 Phone: 345-333-543 Update Listing	Entertainment

1 2 3 4 5

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы научились различным способам форматирования таблицы данных.

Вы научились:

- Изменять заголовки столбцов.
- Сортировать данные в таблице.
- Добавлять разбиение страниц, чтобы разбить содержимое сайта на несколько более мелких Web-страниц.
- Использовать группы элементов различными способами для изменения разметки таблицы на Web-странице.
- Применять CSS для улучшения внешнего вида Web-страницы.

Урок 2.2. Применение компонента загрузки файлов

При добавлении столбца ФОТОГРАФИЯ необходимо предоставить пользователям возможность загружать изображения для объявлений. Особенно важна эта функция для страниц `new_record.jsp` и `update_record.jsp`. Компонент загрузки файлов позволяет пользователям просматривать файловые системы, загружать файлы в базу данных и немедленно отображать эти файлы в базе данных.

Так как процедура использования компонента загрузки файлов схожа для процедуры создания и обновления объявлений, данная процедура подойдет для обновления страницы с учетом текущего изображения. При необходимости можно внести некоторые изменения в страницу создания новой записи после обновления страницы.

Добавьте текущую фотографию на страницу

Вы заметите, что форма обновления является обычной таблицей HTML со статичным текстом и некоторыми компонентами ввода, привязанными к столбцам в реляционной записи `update_record`. Учитывая это можно добавить строки и столбцы для изменения формы, точно так же, как и при работе с любой другой таблицей HTML. В последующих шагах вы добавите новую строку для отображения текущей фотографии.

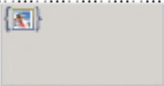
Добавление строк и столбцов сложнее в таблице данных, отображающей список реляционных записей. Информация о добавлении столбцов к таблице данных приведена в разделе “Урок 1.2. Подключение к базе данных и отображение данных на Web-странице” на стр. 6.

1. Дважды щелкните на странице **update_record.jsp** в панели Меню проектов.
2. Поместите курсор в первую (левую верхнюю) ячейку таблицы. Эта ячейка помечена как ИД.
3. В строке меню выберите **Таблица → Добавить строку → Добавить выше**, чтобы добавить новую строку вверху таблицы, предназначенную для текущей фотографии для записи.
4. В крайней левой ячейке новой строки введите **Текущая фотография:**, что будет играть роль метки.
5. Перенесите компонент **Изображение** из блока Компоненты Faces Палитры в крайнюю правую ячейку новой строки. В отличие от предыдущего упражнения, вы позволите пользователю задать полноразмерное изображение без ограничения размеров, поэтому не изменяйте значения ширины и высоты в панели Свойства.
6. Привяжите компонент изображения к столбцу ФОТОГРАФИЯ для `update_record`, переместив столбец ФОТОГРАФИЯ из панели Данные страницы в новый компонент изображения. Компонент изображения теперь привязан к столбцу ФОТОГРАФИЯ базы данных.
7. Оставив графическое изображение выбранным, перейдите в панель Свойства и нажмите кнопку **Выбрать объект данных страницы** рядом с полем Тип. Откроется окно **Выбрать объект данных страницы**.
8. Щелкните на значке +, чтобы развернуть реляционную запись `update_record`.
9. Выберите **ТИП ИЗОБРАЖЕНИЯ (String)**.
10. Нажмите кнопку **ОК**, на странице будет показана текущая фотография для объявления. Страница будет выглядеть следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#)

[Post New Listing](#)

Current photo:	
Id	(ID) 123
Title:	{TITLE} abc
Description:	{DESCRIPTION} abc
Maincategory:	{MAINCATEGORY} abc
Price:	{PRICE} 123
Phone:	{PHONE} abc
{Error Messages} ↓	
Update Delete	

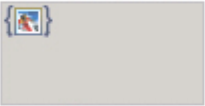
Добавьте компонент загрузки файла на страницу

Затем добавьте новую строку внизу таблицы для хранения компонента загрузки файлов.

1. Поместите курсор в последнюю строку, щелкнув внутри ячейки с текстом "Телефон:".
2. В строке меню выберите **Таблица** → **Добавить строку** → **Добавить ниже**.
3. В первой ячейке новой строки введите в качестве метки текст **Создать фотографию:**.
4. Перенесите компонент **Загрузка файла** из блока **Расширенные компоненты Faces** Палитры в последнюю ячейку новой строки.
5. Так же, как вы привязали компонент изображения, привяжите компонент загрузки файлов к столбцу **ФОТОГРАФИЯ** реляционной записи `update_record`. В текстовом поле будет показано **{ФОТОГРАФИЯ}**, чтобы указать, что данный компонент привязан к столбцу **ФОТОГРАФИЯ** и что загружаемый файл будет помещен в этот столбец. Ваша страница будет выглядеть следующим образом:

Welcome to the Classifieds!

[View All Classifieds](#)[Post New Listing](#)

Current Photo:	
Id:	{ID} 123
Title:	{TITLE} abc
Description:	{DESCRIPTION} abc
Maincategory:	{MAINCATEGORY} abc
Price:	{PRICE} 123
Phone:	{PHONE} abc
New Photo:	{PHOTO} Browse...
{Error Messages} ↓	
Update Delete	

6. Сохраните страницу и при необходимости проверьте ее.

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы научились работать с таблицей на странице обновления, добавлять текущую фотографию в таблицу и добавлять компонент загрузки файлов в таблицу.

Урок 2.3. Применение правил навигации

Способ настройки страницы new_record.jsp заставляет пользователя постоянно следить за тем, чтобы не ввести уже используемый числовой ИД, поскольку у каждой записи в базе данных должен быть уникальный числовой ИД. Подробно это было объяснено в разделе “Урок 1.4. Создание записи для отображения и обновления базы данных” на стр. 14. Если коротко, то вы должны сделать так, чтобы пользователь не мог совершить эту ошибку.

В следующем упражнении, “Урок 2.4. Применение автоматической генерации ключей” на стр. 45, вы воспользуетесь функцией автоматической генерации ключей, устраняющей необходимость ввода числового ИД. В этом упражнении вы проверите, уникален ли введенный ИД, и если нет, то вы отправите пользователя на страницу ошибок, содержащую описание неполадки и инструкции по ее устранению.

Правила навигации позволяют направить пользователя на страницу ошибок или на страницу all_records.jsp в зависимости от результатов проверки уникальности введенного пользователем числового ИД. Вы присвоите

псевдонимы двум возможным результатам такой проверки и затем свяжете эти псевдонимы с соответствующими целевыми страницами. В данном примере ошибке на странице `create_record.jsp` будет соответствовать псевдоним `ERROR_CREATE`, который отправит пользователя на страницу ошибок. Правильное заполнение страницы `create_record.jsp` будет обозначаться псевдонимом `MAIN`, который будет связан со страницей `all_records.jsp`, как обычно.

Настройте правила

1. Дважды щелкните на странице **`new_record.jsp`** в Навигаторе проектов.
2. Нажмите кнопку **Опубликовать новый список** на странице.
3. В панели Свойства нажмите **Добавить правило**. Откроется окно Добавить правило навигации. Первое правило будет направлять пользователя на страницу ошибок `create_error.jsp`, если во время создания и занесения новой записи в базу данных возникнет ошибка.
4. В поле **Страница** введите `/error_create.jsp`. Эта страница не существует, но для целей учебника можно представить, что она есть.
5. Отметьте радиокнопку **Результат с именем**.
6. Введите `ERROR_CREATE` в текстовом поле после радиокнопки **Результат с именем**.
7. Отметьте радиокнопку **Только эта страница**, поскольку на сайте нет других страниц, на которых может возникнуть данная ошибка при вводе пользователем неunikального числового ИД. Нажмите **ОК**. Следующее правило будет перемещать к странице `all_records.jsp`, если пользователь ввел допустимый числовой ИД.
8. Нажмите кнопку **Добавить правило**, чтобы вновь открыть окно Добавить правило навигации.
9. В выпадающем списке **Страница** выберите **`all_records.jsp`**.
10. Отметьте переключатель **Результат с именем** и затем введите `MAIN` в расположенном рядом с ним текстовом поле.
11. Поскольку это правило может понадобиться вам и на другой странице (например, на странице `update_record.jsp`), отметьте радиокнопку **Все страницы** рядом с полем **Это правило используется для**. Нажмите кнопку **ОК**. Два указанных правила теперь показаны в панели Свойства.

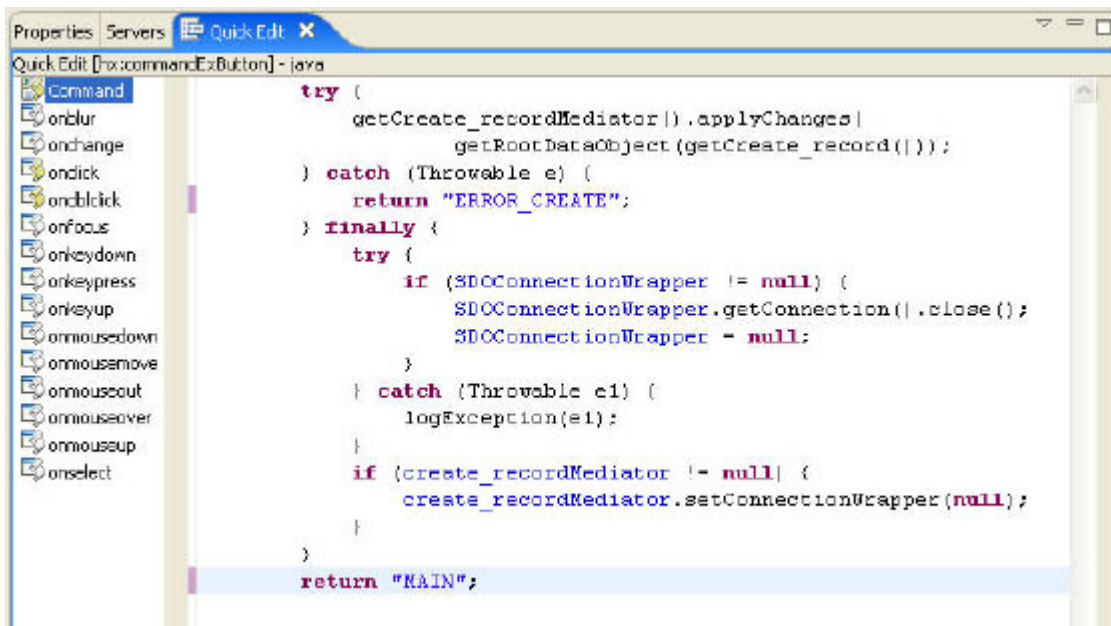
Действие Возвращать псевдонимы по кнопке

Все, что осталось, - ввести новые правила навигации в действие. Вы добавите два оператора `return` в код кнопки Опубликовать новый список. Эти операторы `return` вызывают соответствующие псевдонимы, в результате чего пользователь попадает на страницу, заданную в правиле навигации.

1. Нажмите кнопку **Опубликовать новый список**.
2. Откройте панель **Быстрое редактирование**.
3. В панели Быстрое редактирование найдите строку `} catch (Throwable e) {`. Эта функция `catch` запускается, если пользователь введет неunikальный числовой ИД.
4. Удалите весь код между открывающей фигурной скобкой `{` в конце этой строки и следующей закрывающей фигурной скобкой `}` несколькими строками ниже. Не удаляйте ни ту, ни другую фигурную скобку.
5. Вместо кода между фигурными скобками введите следующий текст: `return "ERROR_CREATE";`
6. Затем удалите весь код после последней закрывающей фигурной скобки `}` и вместо него введите следующий текст: `return "MAIN";`

Это удалит действие `gotoPage`, добавленное вами при выполнении упражнения 1.4. Вам больше не нужен этот код, поскольку правила навигации делают то же самое.

Код кнопки теперь будет выглядеть следующим образом:



7. Сохраните страницу и при необходимости проверьте ее.

При желании вы можете создать простую страницу ошибок create_error.jsp, сообщающую пользователю, что при его попытке создать список произошла ошибка и ему следует указать другой ИД. Затем вы можете проверить эти правила навигации, попытавшись добавить новый список, использующий существующий ИД (например, 1).

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы научились применять правила навигации, создающие ошибку, если пользователь пытается ввести недопустимый ИД, или направляющие пользователя на страницу all_recordlist.jsp, если пользователь вводит допустимый ИД.

Урок 2.4. Применение автоматической генерации ключей

Лучший способ задавать для каждой записи уникальный ИД - сделать так, чтобы база данных автоматически присваивала уникальный номер каждой записи. В данном упражнении вы настроите автоматическую генерацию ключей, чтобы сайт автоматически присваивал новый номер каждой новой записи базы данных. Автоматическая генерация ключей - это сложный вопрос. Если коротко, то база данных может выбирать новые ключи при наличии специальной таблицы, зарезервированной для генерации ключей. В этой таблице должен быть список неиспользуемых ключей в одном столбце (столбце приращения) и список чисел, начиная с 1, во втором столбце (столбце идентификаторов). Когда базе данных потребуется новый ключ, она извлечет его из строки с номером 1 в столбце идентификаторов и затем получит новый ключ, готовый для следующего раза.

В предыдущем упражнении вы настроили правила навигации, контролирующие уникальность введенного пользователем ИД для нового объявления. Такая процедура неудобна, поскольку требует, чтобы пользователь угадывал ключи.

Конечно, лучший способ задавать для каждой записи уникальный ИД - сделать так, чтобы база данных автоматически присваивала уникальный номер каждой записи. В данном упражнении вы настроите автоматическую генерацию ключей, чтобы сайт автоматически присваивал новый номер каждой новой записи базы данных.

Автоматическая генерация ключей - это сложный вопрос. Если коротко, то база данных может выбирать новые ключи при наличии специальной таблицы, зарезервированной для генерации ключей. В этой таблице должен быть список неиспользуемых ключей в одном столбце (столбце *приращения*) и список чисел, начиная

с 1, во втором столбце (столбце *идентификаторов*). Когда базе данных потребуется новый ключ, она извлечет его из строки с номером 1 в столбце идентификаторов и затем получит новый ключ, готовый для следующего раза.

Дополнительные сведения об автоматической генерации ключей:

Для того чтобы вы могли настроить автоматическую генерацию ключей, вы должны знать, как она устроена. Для применения автоматической генерации ключей необходимо, чтобы в базе данных существовала таблица, специально предназначенная для этой цели. Эта таблица состоит из двух столбцов:

- Столбец *приращения* хранит доступные ключи. Когда базе данных требуется новый уникальный ключ, она извлекает его из этого столбца.
- Столбец *идентификаторов* содержит перечень чисел, в котором есть только одна единица. Этот столбец указывает базе данных, какой ключ следует выбрать в столбце приращения. Столбец идентификаторов - это главный ключ таблицы генерации ключей.

Когда базе данных требуется новый уникальный ключ, она находит в столбце идентификаторов строку с единицей. Значение столбца приращения для этой строки дает следующий доступный ключ. С помощью этого ключа база данных обновляет таблицу, так что в следующий раз будет доступен новый ключ.

Ниже приведен пример таблицы генерации ключей. Каков следующий доступный ключ для этой базы данных? Ответ указан под таблицей.

Таблица 1. Таблица генерации ключей

Столбец идентификаторов	Столбец приращения
3	78
4	3
1	65
2	12

Следующий доступный ключ в этой таблице - 65, поскольку именно этот ключ (в столбце приращения) находится в той же строке, что и 1 - в столбце идентификаторов.

После извлечения очередного доступного ключа из таблицы она обновляется для следующего случая, когда понадобится ключ. База данных может извлекать и по несколько ключей за раз, используя значения из нескольких строк столбца приращения.

Вкратце, для применения автоматической генерации ключей необходима лишь таблица генерации ключей, состоящая из двух столбцов: столбца главного ключа, используемого в качестве столбца идентификаторов, и столбца для хранения следующего доступного ключа. При инициализации в эту таблицу должна быть занесена одна запись, у которой в столбце идентификаторов находится единица, а в столбце приращения - первый доступный ключ. После настройки таблицы можно начинать пользоваться автоматической генерацией ключей.

Настройте автоматическую генерацию ключей

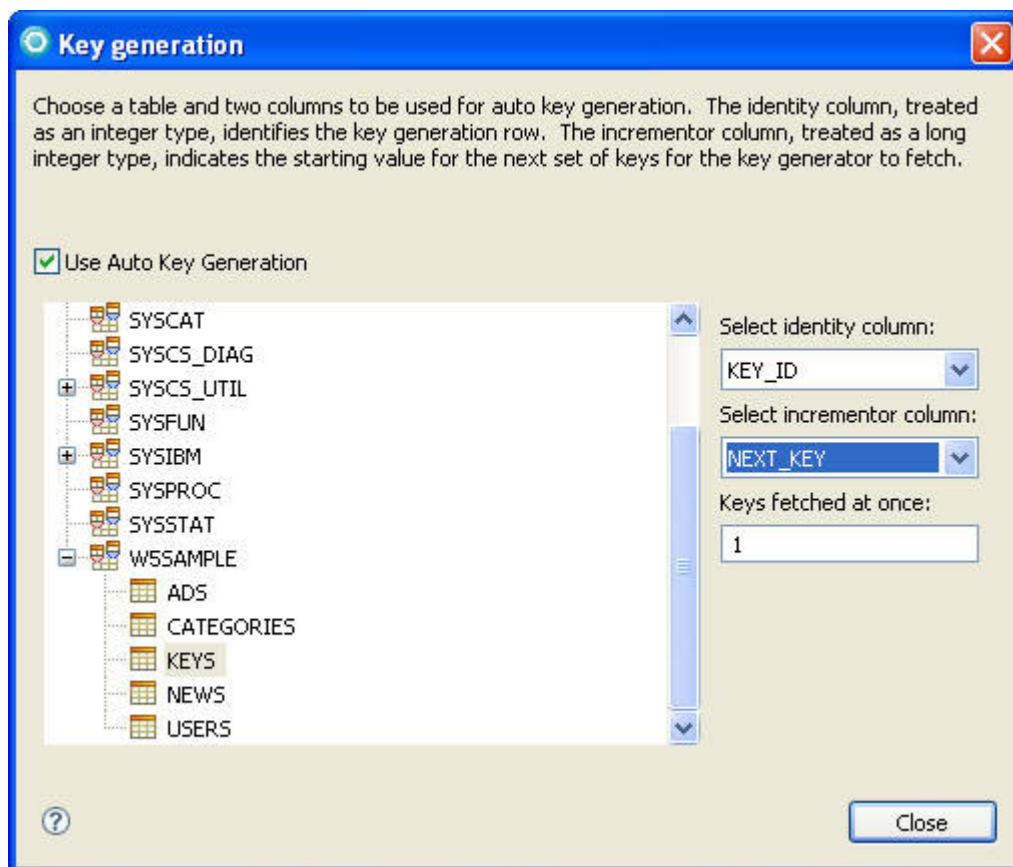
В примере базы данных для данного учебника имеется таблица генерации ключей с именем KEYS. В двух ее столбцах, как описано выше, содержатся новые ИД для всех записей. В данных шагах вы настроите реляционную запись `create_record` для извлечения ИД из таблицы KEYS.

1. Дважды щелкните на странице `new_record.jsp` в панели Меню проектов.
2. Дважды щелкните на реляционной записи `create_record` в панели Данные страницы. Появится окно Настроить реляционную запись.
3. Отметьте переключатель "Повторно использовать определение метаданных из существующей записи или списка записей" и радиокнопку "Заполнять запись существующими данными из базы данных", затем нажмите **Далее**.

4. Выберите соединение с базой данных в выпадающем списке, затем нажмите **Далее**.
5. В правой части окна Настроить реляционную запись приведен список ссылок. Выберите **Автоматическая генерация ключей**. Откроется окно Генерация ключей.
6. Отметьте переключатель рядом с **Использовать автоматическую генерацию ключей**.
7. Разверните список W5SAMPLE, затем выберите таблицу **W5SAMPLE.KEYS**.
8. В поле **Выбрать столбец идентификаторов** укажите **KEY_ID**.
9. В поле **Выбрать столбец приращения** укажите **NEXT_KEY**.

Так как для ИД объявления вам понадобится только один ключ, оставьте параметр "Число извлекаемых за раз ключей" равным 1. Если вам понадобится несколько ключей, при таком значении параметра база данных будет выбирать их все за один раз.

Окно Настроить объект данных должно выглядеть следующим образом:



10. Нажмите **Заккрыть**, чтобы закрыть окно Генерация ключей, затем **Готово**, чтобы применить изменения. Теперь поле ИД будет создаваться автоматически для каждой новой записи. Теперь необходимо удалить поле ввода ИД, чтобы пользователь не мог вручную ввести значение.
11. Поместите курсор на верхней строке таблицы формы ввода, щелкнув на тексте **ИД:**.
12. Выберите **Таблица** → **Удалить** → **Удалить строку**.
13. Сохраните страницу.
14. Убедитесь, что функция автоматической генерации ключей правильно добавлена в исходный код страницы:
 - a. Щелкните правой кнопкой мыши на странице **new_record.jsp** в Проектировщике страниц и выберите **Редактировать код страницы**. Откроется файл New_record.java.
 - b. Найдите метод `getCreate_record()`.
 - c. Убедитесь, что перед оператором `catch` добавлена следующая строка кода:
`getCreate_recordMediator().autoGenerateKey(create_record);`

Если вы хотите просмотреть сгенерированный для вас ключ, то можете удалить компонент ввода для данного ИД и заменить его на компонент вывода, привязанный к столбцу ИД для `create_record`. После этого автоматически сгенерированный ключ появится вверху формы, но будет недоступен пользователю.

Проверьте выполненные задания учебника

Когда ваше Web-приложение будет готово к опубликованию, вам потребуется сервер для хостинга, чтобы пользователи смогли обращаться к Web-сайту через Internet. Однако для тестирования Web-сайта можно воспользоваться WebSphere Application Server для имитации сервера. Инструкции по тестированию Web-сайта приведены в разделе “Урок 1.3. Тестирование Web-сайта” на стр. 12.

Контрольные точки урока

Вы закончили выполнение этого урока. В этом уроке вы научились настраивать автоматическую генерацию ключей.

Модуль 2: итоги

Вы закончили работу с модулем 2, Добавление дополнительных функций.

Пройденные уроки

Теперь вы умеете:

- Форматировать записи базы данных на Web-странице
- Добавлять компонент, позволяющий загружать файлы в базу данных из Web-страницы
- Автоматически переходить с одной страницы на другую
- Автоматизировать выполнение некоторых задач администрирования базы данных

Дополнительные ресурсы

Ниже приведены ссылки на разделы с дополнительной информацией об этом продукте или о JavaServer Faces и технологии JSF.

Связанная информация

ibm.com

eclipse.org

Технология JavaServer Faces

JSF в developerWorks

Отображение динамических сведений на Web-страницах с JavaServer Faces

Вы закончили работу с учебником Отображение динамических сведений на Web-страницах с JavaServer Faces. Из него вы узнали, как использовать JavaServer Faces для создания Web-страниц, способных извлекать информацию из базы данных.

Пройденные уроки

Из данного учебника вы узнали о следующих концепциях и задачах:

- Как подключать Web-страницы к базе данных.
- Как создавать страницы, способные создавать, считывать, обновлять и удалять записи из базы данных.
- Как отправлять данные из одной страницы на другую.
- Как форматировать динамическое содержимое Web-страниц.
- Что такое JavaServer Faces и компоненты JSF

Дополнительные ресурсы

Дополнительные сведения по темам, рассмотренным в данном учебнике, приведены в следующих источниках:

Связанная информация

Технология JavaServer Faces

JSF в developerWorks