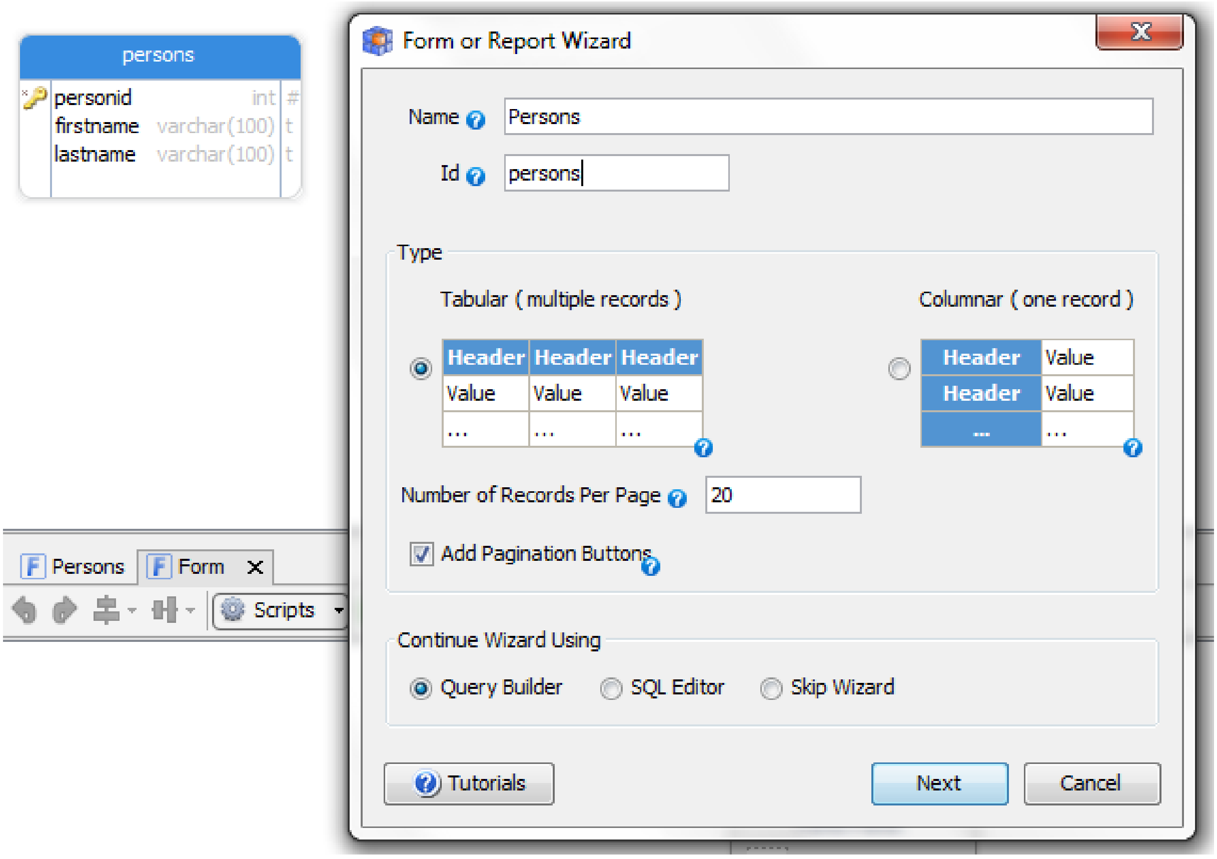
DbSchema Forms and Reports

Use DbSchema forms and reports designer to build small applications or reports for web or Swing. Take the advantage of:

- Interactive forms, with dialogs, buttons, charts
- Form applications can be deployed as Tomcat WAR
- Use of responsive bootstrap HTML templates for correct display on mobile devices
- Unlimited deepness of master-details in reports
- Use of JavaScript or Groovy scripts

Create First Form Using the Wizard

First please connect to a database (can be any database). In this sample we will connect to a MySql database. There please create a table Persons with the columns personid, firstname, lastname. Right-click the table header and choose ‘Random Data Generator’ to fill the table with random data. Then right-click again the table header and choose ‘Form or Report’ which will start the Forms Wizard.



persons

personid	int	#
firstname	varchar(100)	t
lastname	varchar(100)	t

Form or Report Wizard

Name: Persons

Id: persons

Type

● Tabular (multiple records) ○ Columnar (one record)

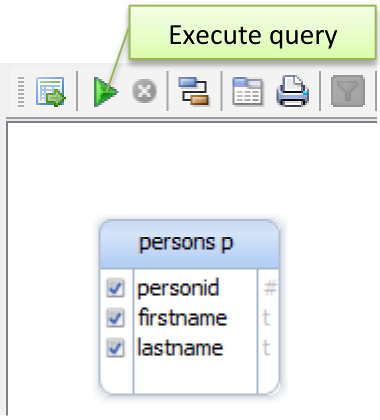
Number of Records Per Page: 20

☑ Add Pagination Buttons

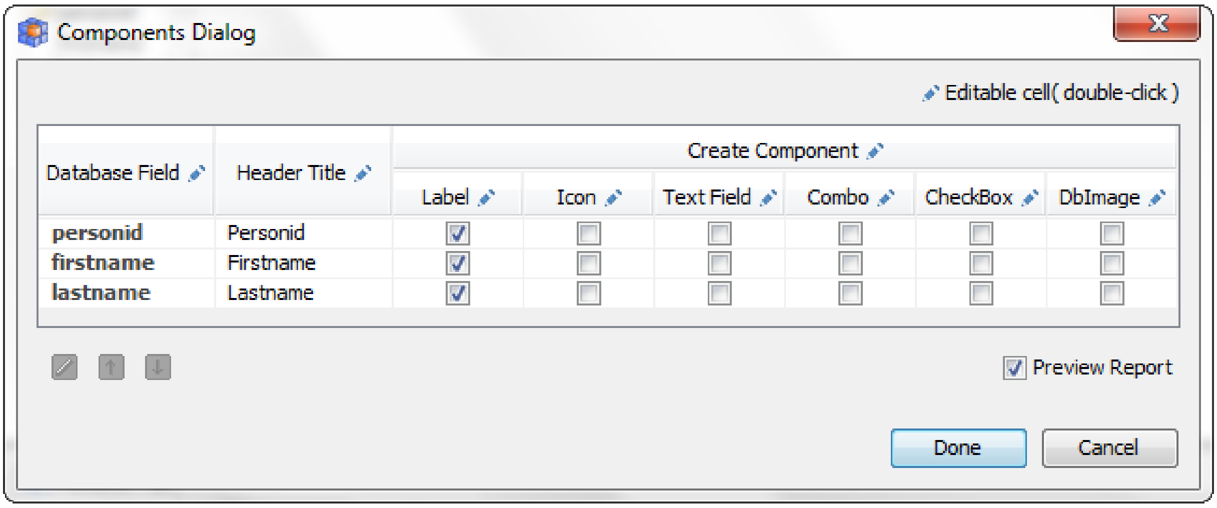
Continue Wizard Using

● Query Builder ○ SQL Editor ○ Skip Wizard

In the Forms wizard keep the default settings and press ‘next’. The Query builder will show. Ensure all columns are ticked and press ‘Run’ and ‘Next’.



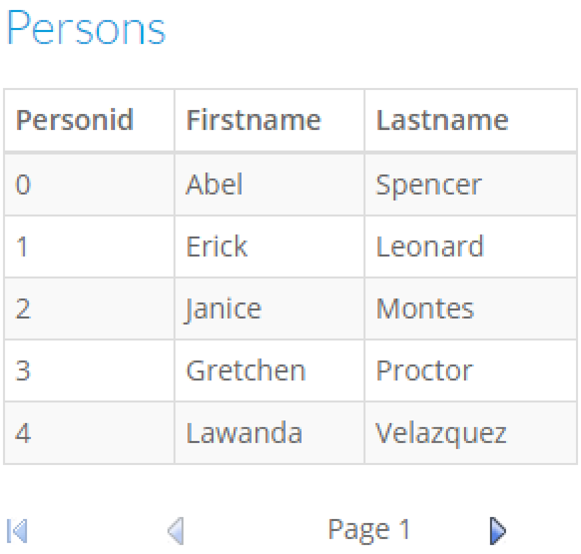
The next dialog will let us choose which fields to place in the report. We will use labels for all fields.



Database Field	Header Title	Create Component					
		Label	Icon	Text Field	Combo	CheckBox	DbImage
personid	Personid	☑	□	□	□	□	□
firstname	Firstname	☑	□	□	□	□	□
lastname	Lastname	☑	□	□	□	□	□

☑ Preview Report

Press ‘Run’ in the created Form designer and you should see this in the Web browser:



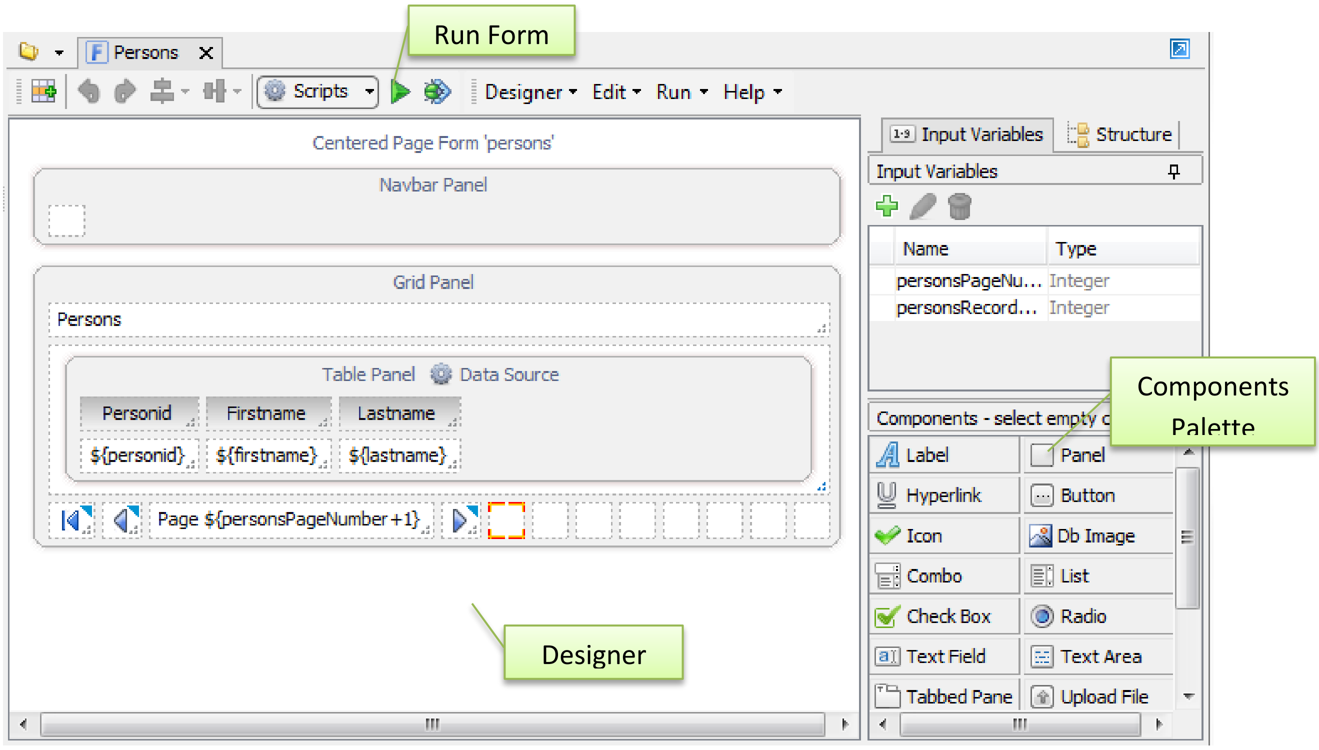
Persons

Personid	Firstname	Lastname
0	Abel	Spencer
1	Erick	Leonard
2	Janice	Montes
3	Gretchen	Proctor
4	Lawanda	Velazquez

Page 1

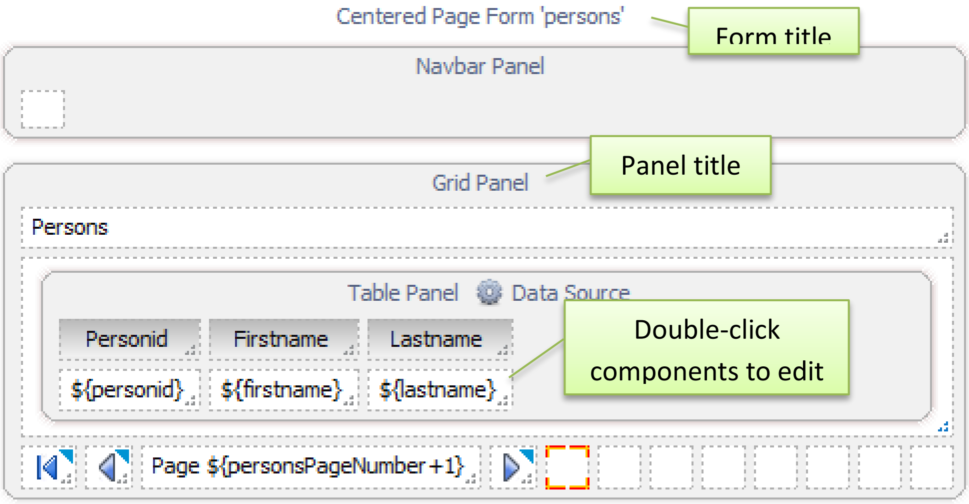
Understand the Designer

The Forms designer looks like this:

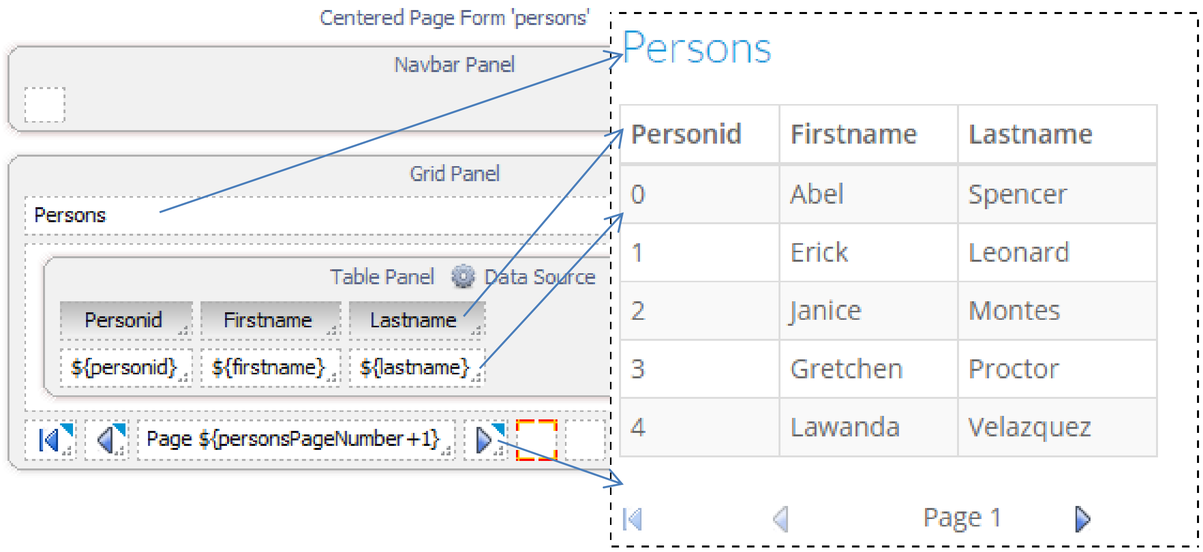


In center is the designer, on the rights is the Components Palette (will activate when you select an empty cell in the designer). The designer consists of panels (rounded rectangles) with cells (dotted rectangles). The upper 'navbar' panel is used for menus. The 'grid' panel is the form content. It is called 'grid' because it gets represented as a bootstrap grid, a dedicated structure of <*div*> tags with 12 columns. This panel contains one more panel which will be represented as a table.
As you can see, panels can be placed one inside another, building a hierarchical structure.

Panel properties can be edited by clicking the panel title and components by double-clicking the text.



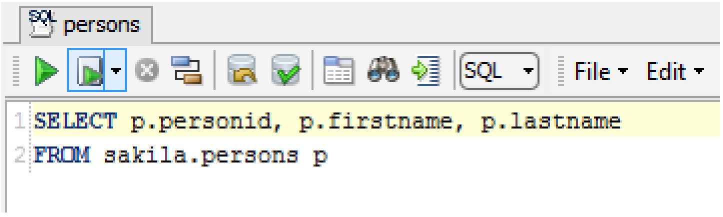
The correspondence between what we see here and the generated output:



Personid	Firstname	Lastname
0	Abel	Spencer
1	Erick	Leonard
2	Janice	Montes
3	Gretchen	Proctor
4	Lawanda	Velazquez

The first component in the grid is a label 'Persons' the title in the report. The grid panel is the entire page body. Inside it is imbricated a panel which is represented as HTML table. The first row in this panel is the table header; the second row will display the panel data.

As you can see, this panel has a Data Source as SQL query. You can view the query by clicking the 'Data Source' text in the panel title.

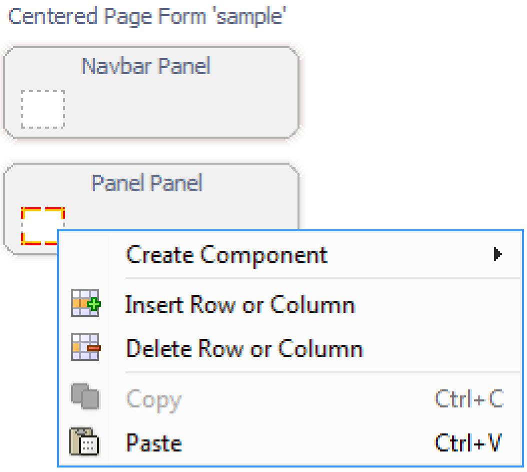


```
SELECT p.personid, p.firstname, p.lastname
FROM sakila.persons p
```

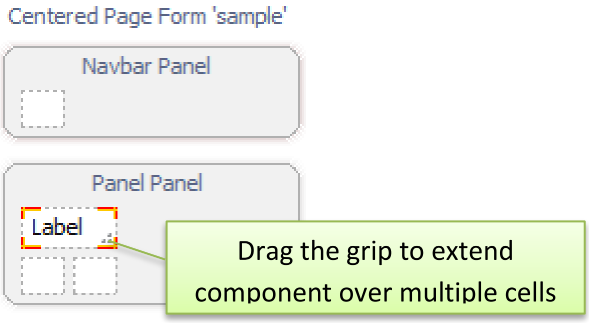
Create and Position Components

Components can be placed inside empty cells. To demonstrate how to place them please create an empty form from the application main menu Forms / Form or Report. In the Forms Wizard choose 'Skip Wizard' and disable pagination buttons.

In the new created form right-click a cell and choose 'Insert Row or Column'. Add one column to the right and one row on the bottom.

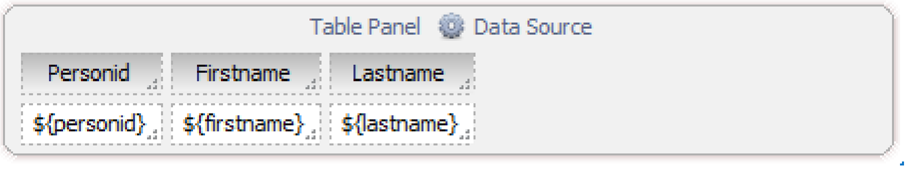


Click the ‘Label’ in the Components palette to create a label in the selected cell. Components can be moved to different cells by dragging the component, or can be sized to cover multiple cells by dragging the grip near component.



Drag the grip to extend component over multiple cells

The data source script generates at runtime variables. For each generated record the entire table panel content excepting the header row will be rendered again and again. A label may display any of this variables if its text contains a ${variableName}, for example ${firstname}. This gives the flexibility to write in a single label even a combination of two variables Example: ‘My name is ${firstname} ${lastname}’. It is also possible to use expressions: ${amount+1}



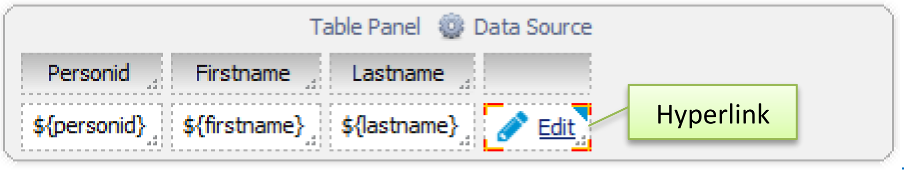
Personid	Firstname	Lastname
${personid}	${firstname}	${lastname}

Each component including panels has a template. For example a label can have H1 template (heading1), H2, PRE, etc. A panel can be a grid (bootstrap grid with 12 columns), table (HTML regular table) or panel (rounded-corner bootstrap panel). Please check http://getbootstrap.com/css/ for understanding them.

Set the component template by double-clicking the component or clicking the title for panels.

Edit Persons data in a Dialog

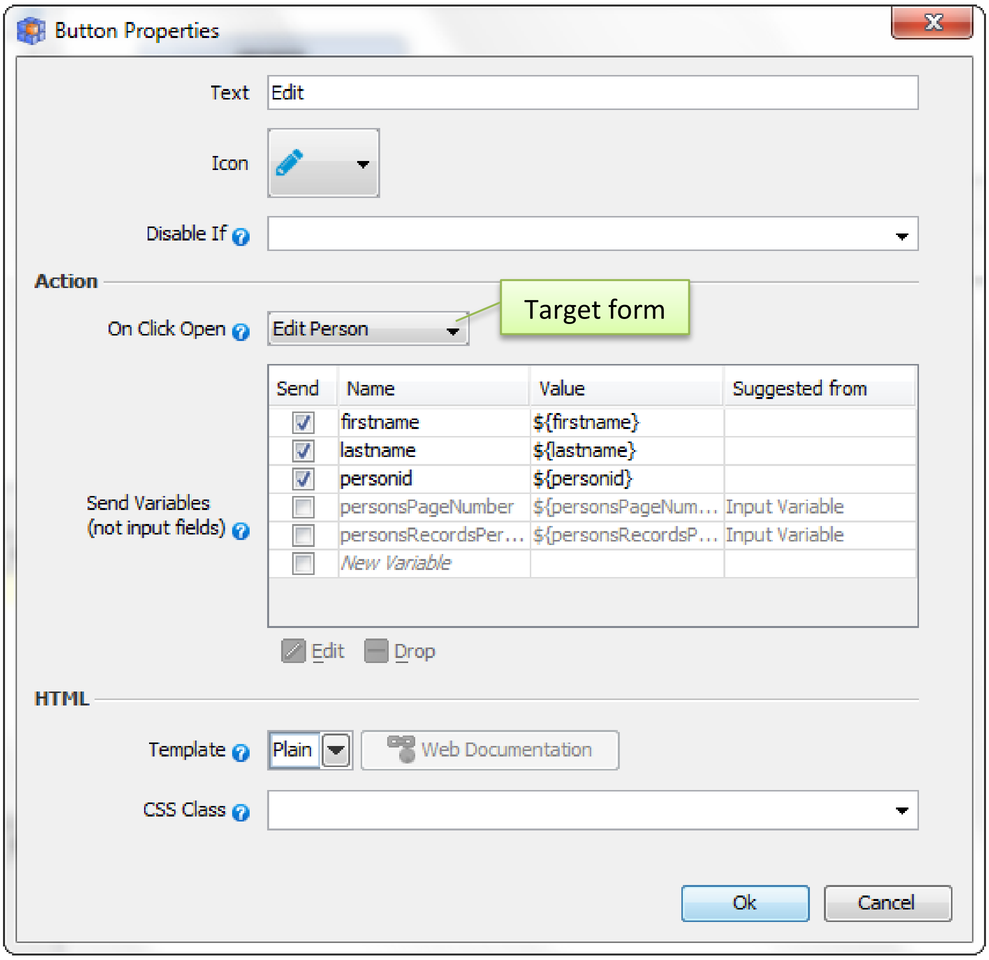
We want to extend our application with a form capable of editing the person data. We will create an ‘Edit Person’ form for editing person firstname and lastname. Therefore we add a hyperlink in the Person form:



Personid	Firstname	Lastname	
${personid}	${firstname}	${lastname}	Edit

Hyperlink

Double-click the hyperlink to edit.



Text: Edit

Action

On Click Open: Edit Person

Target form

Send Variables (not input fields)

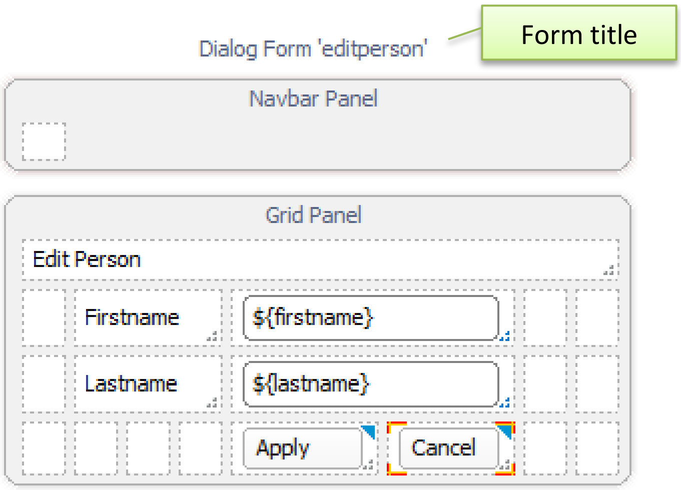
Send	Name	Value	Suggested from
☑	firstname	${firstname}	
☑	lastname	${lastname}	
☑	personid	${personid}	
□	personsPageNumber	${personsPageNum…	Input Variable
□	personsRecordsPer…	${personsRecordsP…	Input Variable
□	*New Variable*		

HTML

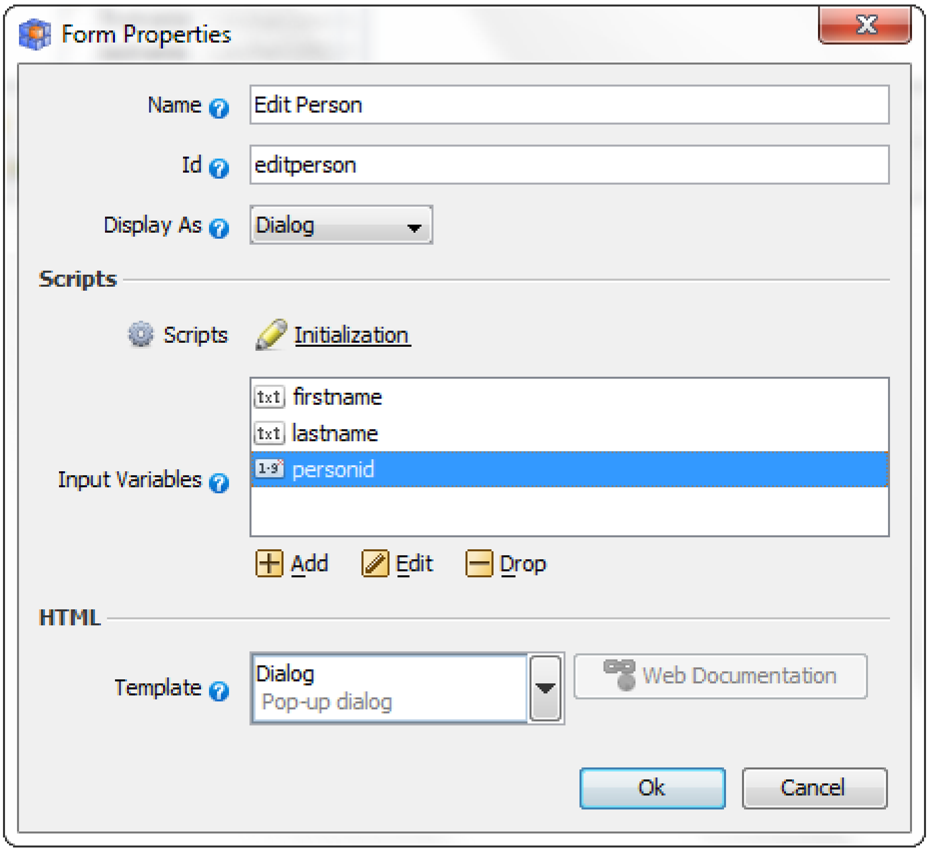
Template: Plain

The hyperlink is set to open the new Edit Person form (here is already created) and send the variables firstname, lastname and personid to it. Many variables are already listed but not ticked, as the designer ‘suggests’ them. They will be sent only after ticking the checkbox. New variables can be added to this list.

Create the Edit Person form as bellow:



Click the form title:



Name: Edit Person

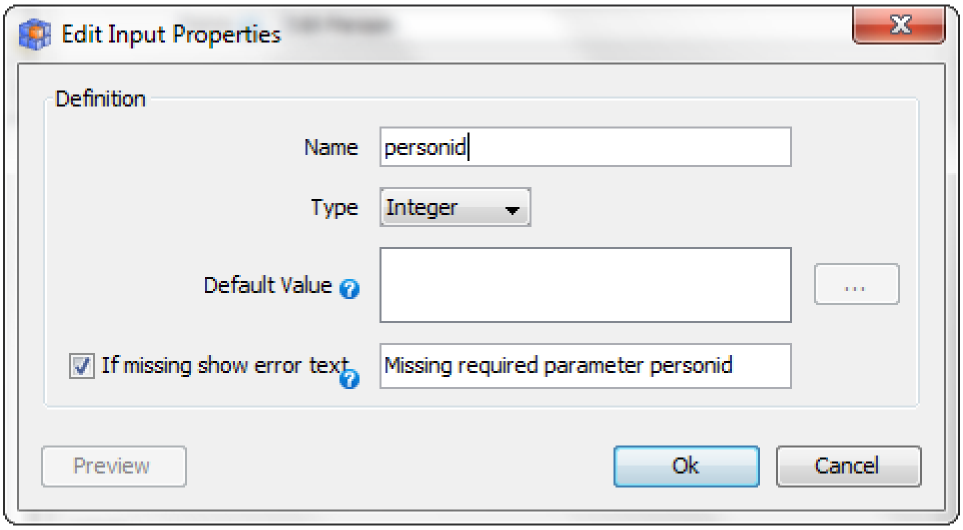
Id: editperson

Display As: Dialog

Input Variables:
- firstname
- lastname
- personid

Template: Dialog
Pop-up dialog

The form is set to show as dialog and the template is also dialog. The input variables which are set are making sure some data is expected. For example personid is set as mandatory input variable and an exception is thrown if personid is missing when opening this form.


Edit Input Properties

Definition

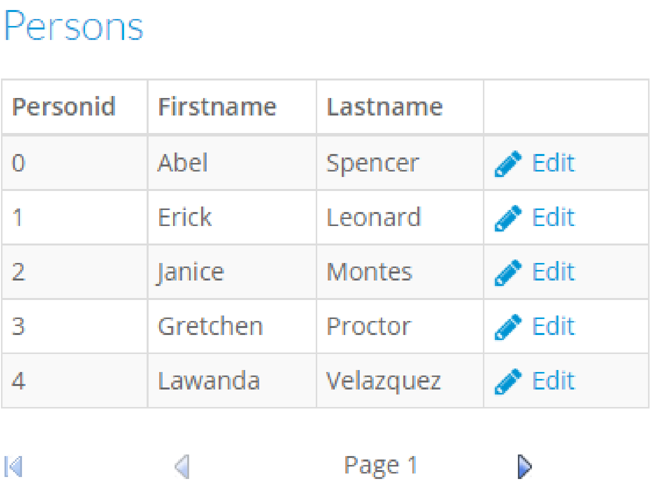
Name: personid

Type: Integer

Default Value:

☑ If missing show error text: Missing required parameter personid

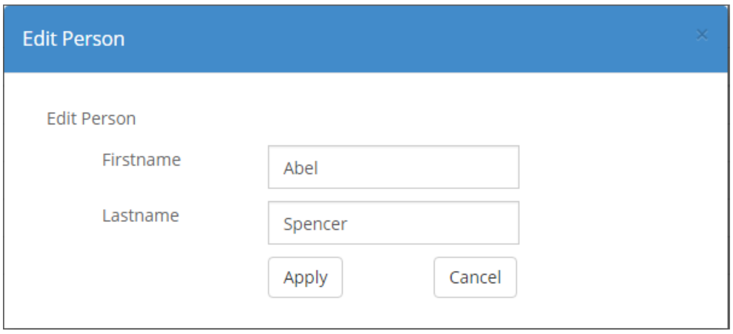
Let’s run the Persons form:


Persons

Personid	Firstname	Lastname	
0	Abel	Spencer	Edit
1	Erick	Leonard	Edit
2	Janice	Montes	Edit
3	Gretchen	Proctor	Edit
4	Lawanda	Velazquez	Edit

Page 1

Clicking the edit will open the Edit Person dialog.


Edit Person

Edit Person

Firstname: Abel

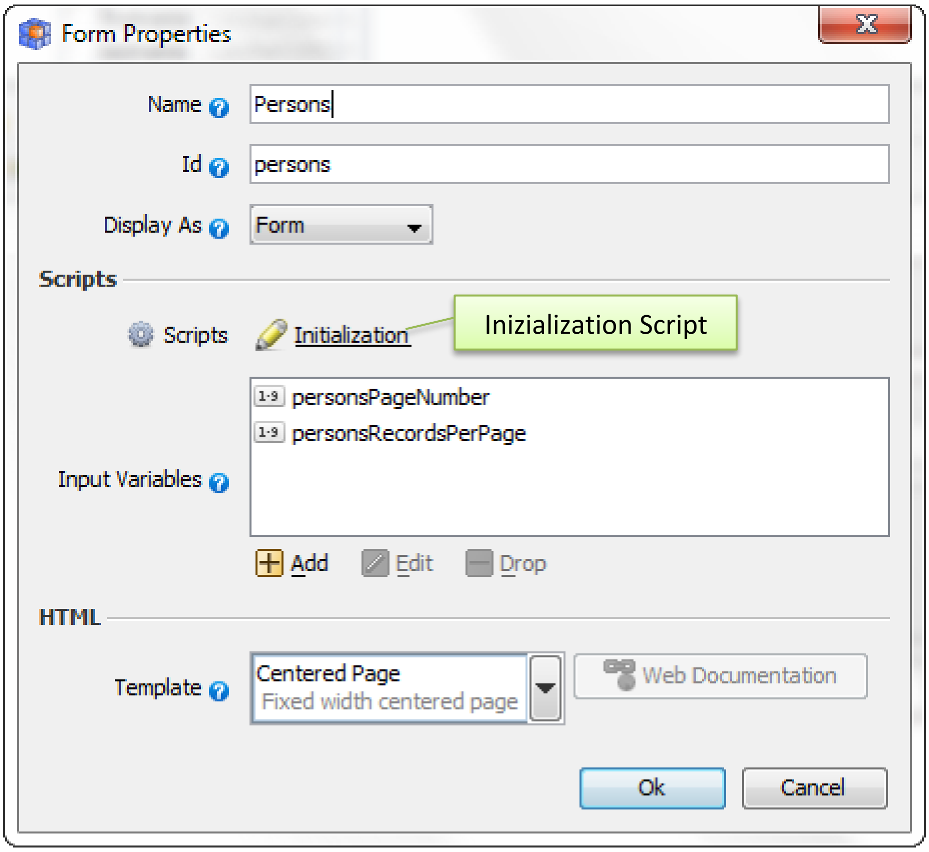
Lastname: Spencer

What we miss now is a script which should modify the data in the database. Where should we create this script? Let’s review the process

- The List Persons will open Edit Persons with personid, firstname, lastname
- Edit Person will call back the List Persons with the personid, edited firstname and lastname. We add here one more variable operation=edit to inform List Person that an editing was done.

- In List Person we may create an Initialization Script (see Form Properties). This script will check for the operation variable and if is equal to 'edit' will modify the data in the database.

Below is the ListPersons form properties dialog where we can enable the Initialization script.



Form Properties

Name: Persons

Id: persons

Display As: Form

Scripts

Scripts: Initialization

Inizialization Script

Input Variables:
- personsPageNumber
- personsRecordsPerPage

HTML

Template: Centered Page — Fixed width centered page

Enter the JavaScript bellow in the editor. The script editor can make use of JavaScript or Groovy. Make sure the language combo is set on JavaScript.

```
print("Operation=" + operation);
 if ( "edit"== operation ){
    print("Updating personid="+personid+"lastname=" + lastname + "
firstname=" + firstname );
    st = sql.prepareStatement("UPDATE persons SET firstname=?, lastname=?
WHERE actorid=?" );
    st.setString(1,firstname);
    st.setString(2,lastname);
    st.setInt(3,actorid);
    st.executeUpdate();
    sql.commit();
 }
```

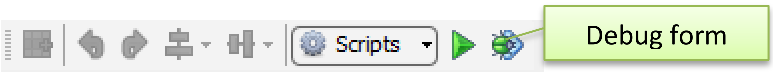
The same script in Groovy is:

```
if ( 'edit'.equals( operation )){
    println "Going to update ${actorid} ${firstname}"
    sql.executeUpdate( "UPDATE persons SET firstname=?, lastname=? WHERE
actorid=?", [firstname, lastname, actorid] )
    sql.commit()
 }
```

We recommend Groovy as being more flexible and powerful. Groovy is also Java based language. In internet there are plenty of Groovy tutorials.

The script contains few print commands used for debugging. Their output will show in the Output logs (Help / Output logs).

Debugging information can get also by running a form in debug mode.



This will show a dialog after clicking any button with the variables which will be send to the target form.

End

The forms and reports designer is new in DbSchema 7. We appreciate any feedback about them, about this document, issues or features.