# Data Center

documentation

Szymon Roszyk and Krzysztof Olejniczak

Poznan 2017

# 1 Problem description

Our goal was to make web application that makes the ability to order the information from data warehouse to customer in user-friendly interface. So far Administrator realized order having manual request from user. Now order is made by service and it is sent to Administrator that can decide which information will be shared with user. The project was commissioned by Roche company and was realized within Data warehouse and analytical processing subject.

Customer does not know what is in the data warehouse. The data in the database for nontechnical user is useless. This is why he cannot use its capabilities properly during reporting. This is the problem that we solved.

Our solution is to create internet service that shows the information about data in understandable way to customer. Using this application user can comfortably show Administrator his needs. Administrator has the ability to manage metadata of the data warehouse and add some information important for nontechnical user. He can add descriptions to columns and tables and hiding them if they are not necessary for customer. Presenting these information our web service uses metadata from data warehouse.

# 2 Requirements

Now we will describe technical requirements of our application and assumptions that we have made.

- Our application needs to be run in a web browser in the client side. It is supported by Firefox 53+, Chrome 58+ and Edge 40+. It also requires enabled JavaScript in browser.
- It's is made in ASP .NET MVC technology so it requires IIS 7.5+ server to run
- It requires Microsoft SQL Server+ to store the program data.
- The application uses also Oracle Database, where database warehouse is stored in. It has the star or snowflake structure. The names of fact tables start with "DM_F_", and the names of dimension tables start with "DM_D_". The application has the permissions to access the metadata USER_OBJECTS, USER_TAB_COLUMNS, USER_CONSTRAINTS of the database.

# 3 Used technologies

Our goal was to make internet application created using ASP .NET MVC 5.0 - a C# framework with HTML 5, CSS 3, Bootstrap 3.3.7, JavaScript, JQuery 3.2.1 and Entity Framework 6.1.3.

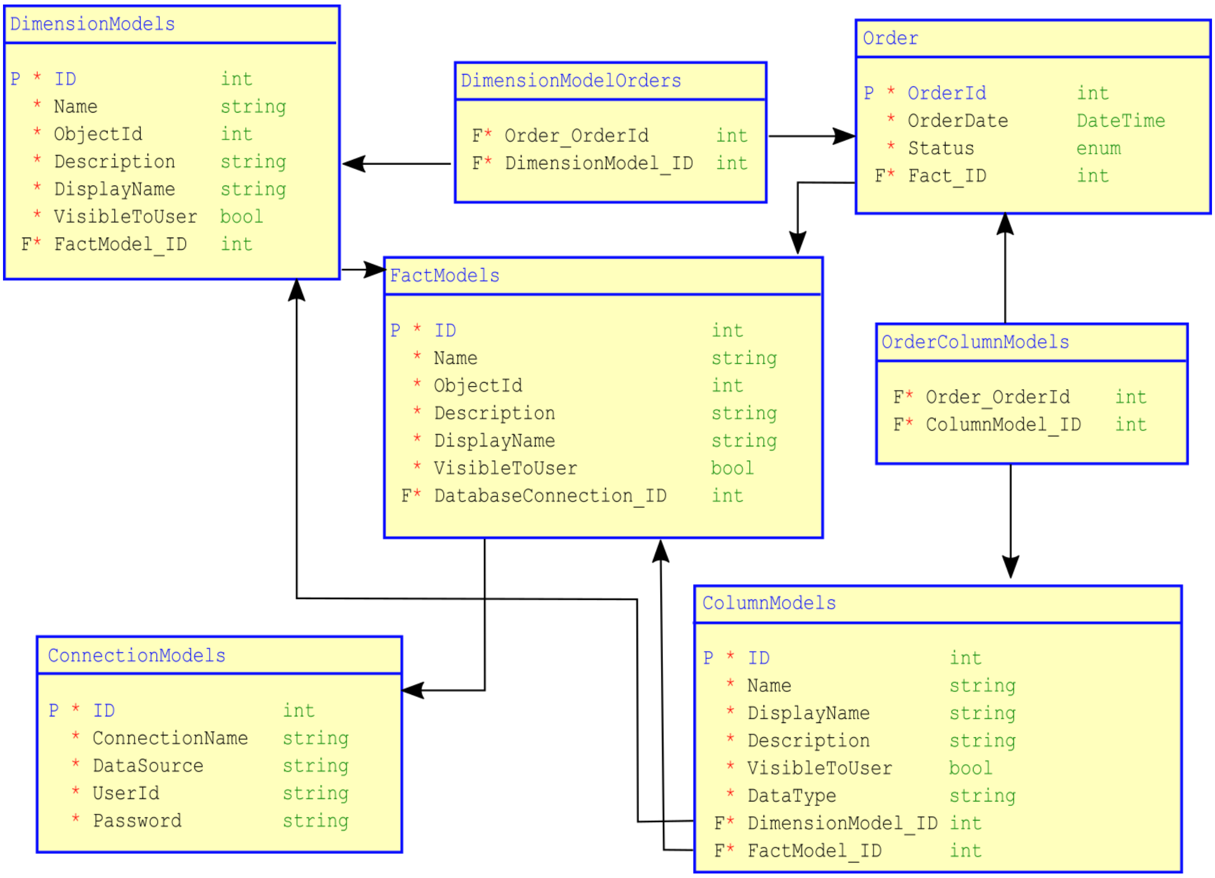
- ASP .NET MVC was used as a back-end to our application. We programmed business logic and database access using it.
- HTML5 was used to show information on web page.

- JavaScript and JQuery was used to change content dynamically especially in editing columns information and preparing the order. JQuery DataTables was used to pagination tables.
- Bootstrap was used to program web site look.
- Entity Framework is an object-relational mapping framework.
- Also we needed Oracle Database Client to connect to Oracle Database from .NET web application.

# 4 Application models

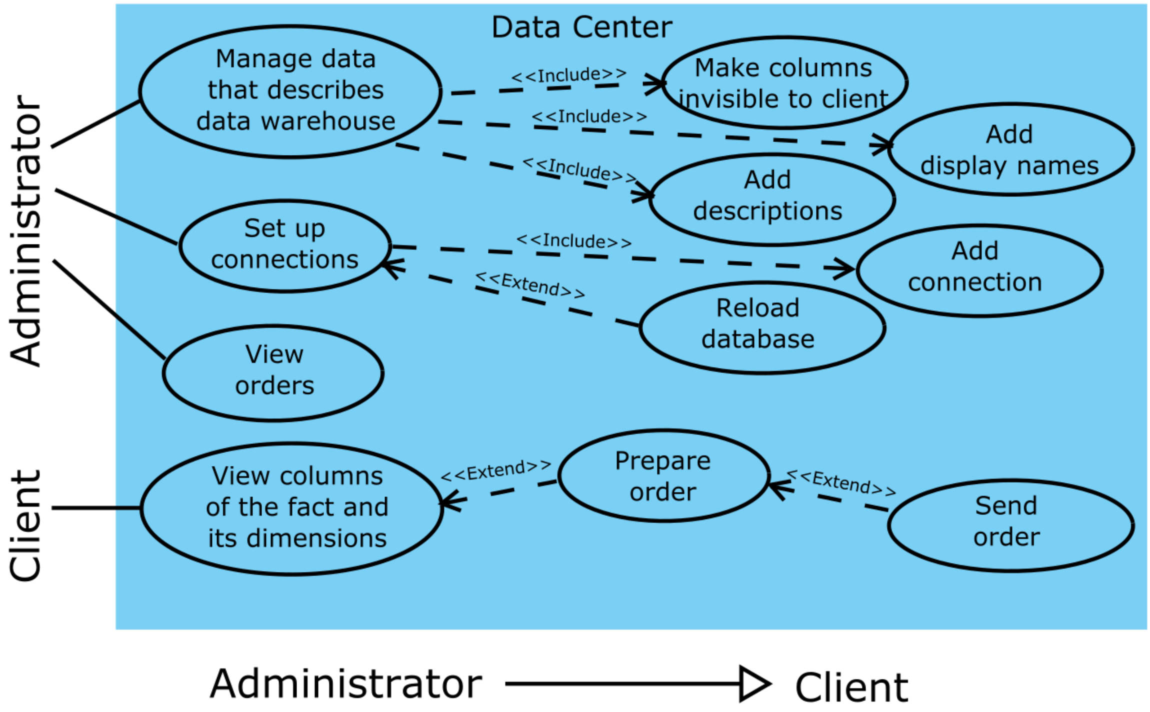
## 4.1 Database model

Below we are presenting our database model that enables program to describe data warehouse. Models are projected into entities in databases using Entity Framework.



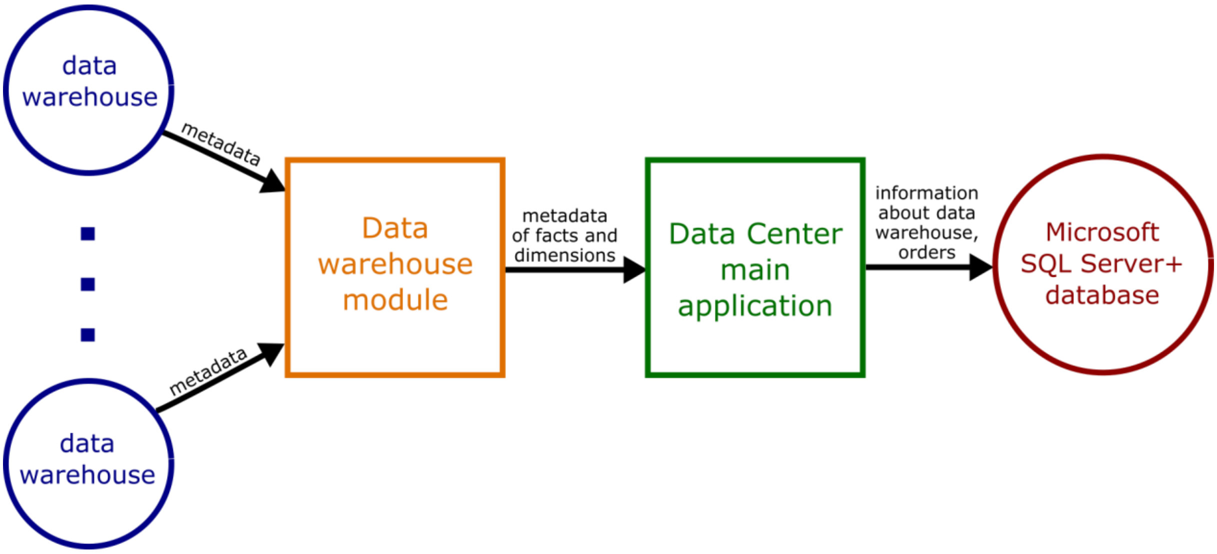
## 4.2 Use cases model

There are roles of the users and actions that they making on our product.



## 4.3 General code structure

In our application there is additional module that enables communication to data warehouse that creates intermediary objects and transmit them into main application, where they are processed into models.


data warehouse
metadata
data warehouse
metadata
Data warehouse module
metadata of facts and dimensions
Data Center main application
information about data warehouse, orders
Microsoft SQL Server+ database

## 5 Database connection modules

Specified connection is transmitted into data warehouse. There is created connection to database and all tables and views available to user are gathered.
Query:

```
SELECT object_name, object_type, object_id
FROM USER_OBJECTS
WHERE object_type IN('TABLE','VIEW')
```

Next for every found table its columns are gathered.
Query:

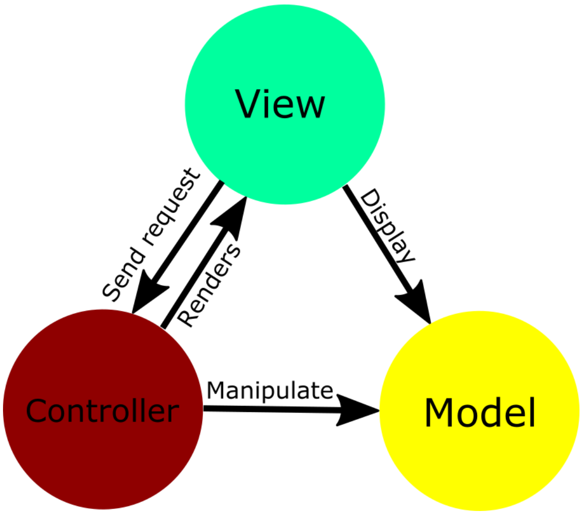
```
SELECT column_name, data_type
FROM USER_TAB_COLUMNS
WHERE table_name = :tableName
```

Collected data are parsed to objects of classes TableObject and ColumnObject.

Tables whose names begin with "DM_F", "V_DM_F" or "S_DM_F" are marked as fact (naming pattern). In database there is a schema of star or snowflake structure so others columns are marked as dimensions. Prepared objects are transmitted into application, where they are transformed into models FactModel, DimensionModel or ColumnModel and saved into database.

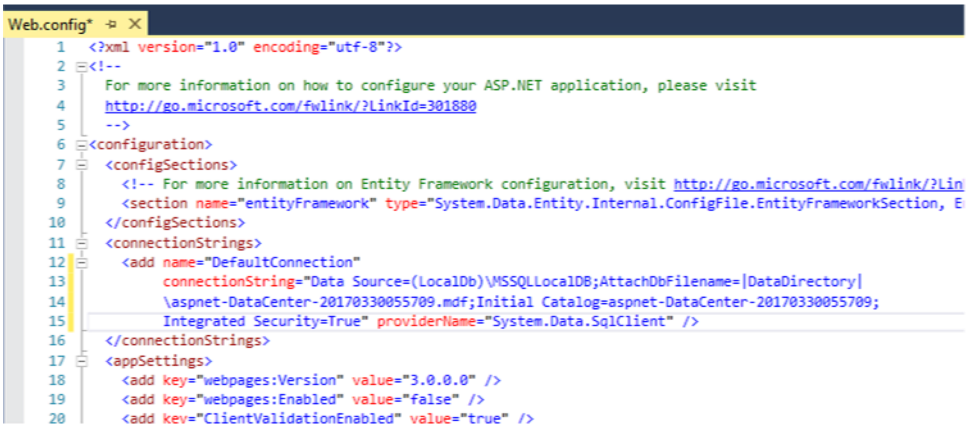
## 6 Application structure

Our application uses model–view–controller pattern to work. It is divided into three parts.



First - Model provide access to database. Classes of model are equivalent to the entities in the database. They are in Model folder.

The information about database connections is stored in web.config.



```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=301880
  -->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?Lin
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, E
  </configSections>
  <connectionStrings>
    <add name="DefaultConnection"
         connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|
         \aspnet-DataCenter-20170330055709.mdf;Initial Catalog=aspnet-DataCenter-20170330055709;
         Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
```

Second - View shows website on screen and allow deciding what to do. They use mix of html5 and C# to represent data on the screen. They are in Views.

Third – Controller connects view and model. They mediate between model and view, and enable making change on data before writing on the screen or writing into database. They are in folder called Controllers.

# 7 Project Components

Now we describe methods and code of classes in our project.

## 7.1 Models

- DataWarehouseContext.cs specifies content of the database. There are five collections representing: Dimensions, Facts, Columns, Connections and Orders.
- AccountViewModels.cs is responsible for storing information about registered users – Administrators.
- ConnectionModel.cs represents connections to the database. It has following fields: ConnectionName, DataSource, UserId and Password. There is also method GetConnectionString that is responsible for convert connection fields to single string needed during connection to database.
- TableModels.cs is class mutual to FactModel and DimensionModel. This class represents mutual fields: ID, Name, ObjectId, Description, DisplayName, VisibleToUser, collections of Columns and Orders. FactModel and DimensionModel are similar, but they have other purposes.
- FactModel.cs specifies single Fact, it inherit from TableModels.cs. It has also list of dimensions and information about connection.
- DimensionModel.cs represents Dimension, it inherit from TableModels.cs. This class has one method – TableObjectsToDimensionsModel that creates list of dimensions.

- ColumnModel.cs stores data about single column in fact or dimension. I has following fields: ID, Name, DisplayName, Description, VisibleToUser, DataType and collection of Orders. There is also one method that provides columns from database.
- Order.cs is responsible for storing orders. The fields are: OrderId, OrderDate, Status, Fact, collections of Dimensions and Colums.

## 7.2 Views

- Shared
  - _Layout.cshtml is responsible for showing main elements of page. It shows bar in a top and links in menu. It also shows footer.
  - _LoginPartial.cshtml is a partial view of _Layout bar on top. It shows login and register button if user is not logged in and it shows hello message and log off if user is logged in.
  - Error.cshtml is a main error page shown in application, it show delivered error message.
  - Lockout.cshtml shows information about locked account. Not used in this application.
- Account
  - Login.cshtml shows form that enables user to log in. It also do initial validation using JavaScript.
  - Register.cshtml is responsible of form that make possible register. It also check if passwords are equal and if they contains proper characters and all fields are filled.
- ConnectionModel
  - Index.cshtml shows list of defined connections. Enables editing and showing details of specified connection. It also make possible to create new connection.
  - Create.cshtml shows form that enables create new connection. It also provides validation.
  - Delete.cshtml is a message that warn against deleting connection. User can confirm delete or resign.
  - Details.cshtml shows details about specified connection.
  - Edit.cshtml enables editing connection. It also validate entered data.
- FactModel
  - Index.cshtml shows list of facts in Administrative tools. It has buttons that enables to show all of columns and dimensions and edit everything about this fact. There are also list of connections and button that enables to reload specified database.
  - Edit.cshtml enables to edit fact. There is ability to set fact or its column or its dimension or its dimension's column as visible, add display name or add description. It uses jQuery to show forms or hide them. It uses partial views to show dimensions, columns and forms.
  - Details.cshtml shows all the details of specified fact: its dimensions, columns and columns of dimensions and their properties.

- DimensionModel
  - _PartialUserDimensionList.cshtml shows list of dimensions to user with display name instead of name where set and description.
  - _PartialDimensionList.cshtml shows list of dimension in administrative tools with information suitable for Administrator.
  - _PartialDimensionEdit.cshtml enables editing dimensions in administrative tools. Every dimension is a link that launches form that enables edit.
- ColumnModel
  - _PartialUserColumnList.cshtml shows list of columns to user with display name instead of name where set, description and checkbox that enables making order.
  - _PartialDimColumnEdit.cshtml enables editing dimension's columns in administrative tools. Every row is a link that launches proper form.
  - _PartialColumnList.cshtml shows the list of columns of table in details view in administrative tools.
  - _PartialColumnEdit.cshtml enables editing fact's columns in administrative tools. Every row is a link that launches proper form.
- Order
  - Index.cshtml shows list of all orders. For Administrator there are buttons that enables showing details of order and change status. User can only see orders, their statues and show details of them.
  - Edit.cshtml shows form that enables changing of status of order.
  - Details.cshtml shows details of order. Firstly it shows fact and after that it shows its columns. Secondarily it shows dimensions and their columns.

## 7.3 Controllers

Now it's time to describe controllers and its methods.

- HomeController.cs is main controller of application.
  - Index gets fact visible to user sends them to view.
  - Show is responsible for view of specified fact and its dimensions. Id of fact is sent by HttpGet, if there is no id or id is incorrect then error appears. Otherwise fact and its dimensions are taken from database and sent into view.
- AccountController.cs performs the user accounts functions. It enables to login and register. It is only useful for administrators because customer doesn't need to log in to make order.
  - Register only views the form that enable to register.
  - HttpPost Register run after clicking "Register" button. It checks correctness of data entered and registers user if they are correct. Otherwise shows error message.
  - Login views the form that make possible to log in to the system.
  - HttpPost Login gets data entered to the form and checks if they are valid. When they are then user is logged in. Otherwise it redirects to previous form and shows error.

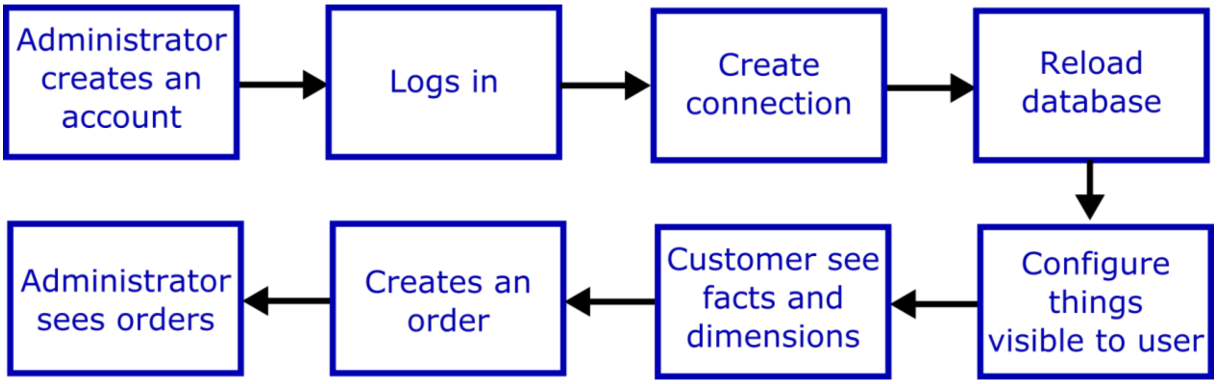
- ConnectionModelController.cs is only for Administrator. It enables to manage connections.
    - Index gets all connection from database and sends them into view.
    - Details gets details about specified connection and sends them into view. Id of connection is sent by HttpGet, if there is no id or id is incorrect then error appears.
    - Create shows form that enable to add new connection
    - HttpPost Create checks correctness of entered data and saves it into database. If data is not correct then it shows the error message.
    - Edit enables to edit specified connection. It gets Id of connection using HttpGet. If there is no Id sent or it is incorrect then error message appears. Otherwise it shows proper form.
    - HttpPost Edit run after clicking "save". It checks correctness of entered data and if they are correct then changes are saved. Otherwise error message appears.
- FactModelController.cs is also only for Administrator. It enables manage facts.
    - Index - using this Administrator can show facts of specified connection given via HttpGet. If there is no connections then empty list is shown. Otherwise list of facts is sent into view.
    - Details – shows details of fact after clicking on them. It gets fact Id via HttpGet. If there is no Id given or given Id is incorrect then error message appears. Otherwise list of columns of the fact and its dimensions are shown.
    - Edit allows setting up display names and description of fact. It takes Id of fact via HttpGet. If There is no Id or Id sent is incorrect then error is shown. Otherwise page that allows editing fact is shown.
    - HttpPost Edit saves changes made in previous method, but before that it checks propriety of entered data. If they are ok then changes are saved. Otherwise error message appears.
    - GetFactColumnForUser gets columns of fact that are visible to user and sends them into view. Id of fact is sent via HttpGet. If there is no Id or it is incorrect then error message is shown.
    - RefreshFacts needs connection string sent via HttpGet. It gets new facts depending on connection and refreshes list of facts. If no connection Id has been sent then error message appears.
- DimensionModelController.cs is responsible for form that enable editing dimension's display name and description. There are also methods that shows dimension and its columns for the user.
    - GetDimensionColumns it sends to view columns of dimension specified by HttpGet Id. If there is not Id specified or Id is incorrect then error appears.
    - GetUserDimensionColumns gets columns from dimension that are visible for user and sends them into view. If given HttpGet Id is not valid then error message is shown.
    - GetDimensionForm enables to show the form of editing specified dimension data. If there is not Id specified or Id is incorrect then error appears.
    - HttpPost Edit cause saving of changes made on GetDimensionForm. If they are not valid then form with error appears.

- ColumnModelController.cs enable edit specific column by Administrator.
  - GetColumnForm enables to edit column of fact. It takes column id and table id via HttpGet and shows error message if that data is incorrect.
  - HttpPost Edit saves changes made on column of fact. If changes are incorrect then shows error.
  - GetDimColumnForm enables to edit column of dimension. It takes column id and table id via HttpGet and shows error message if that data is incorrect.
  - HttpPost EditDimensionColumn saves changes made on column of dimension. If changes are incorrect then shows error.
- OrderController.cs performs operations related to creating and viewing order. Using this controller Administrator has the ability to show and change statuses of orders. User can create order, realize it and show past orders.
  - Index shows all orders made. It is for both Administrator and User, but Administrator can change status.
  - HttpPost Order is responsible for creating order. It is run after clicking "ORDER" button. It gets fact, all dimensions and all columns and save information about them into database.
  - Success shows success message on a screen.
  - Edit enables Administrator to edit status of order. It takes Id via HttpGet and shows error message if Id is bad or not set.
  - HttpPost Edit – save changes made on Edit. Before that checks correctness of that changes.
  - Details shows what exactly was ordered by customer. It shows facts, dimensions and their columns.

# 8 Use of the system

## 8.1 Business process

Here we are presenting business process of application.


Administrator creates an account
Logs in
Create connection
Reload database
Configure things visible to user
Customer see facts and dimensions
Creates an order
Administrator sees orders

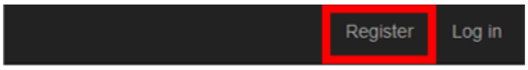
## 8.2 User manual

Now we will describe some instructions of using Data Center system. Every tutorial will start on Home page (HomeController, method index).

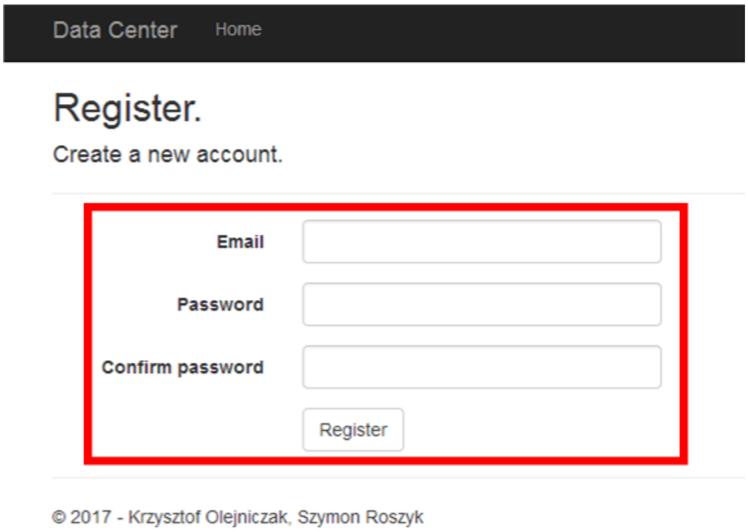
### 8.2.1 Administrative tools

### Register

You have to click "Register" button on right-top side of the page.



Next you have to fill the fields. To "Email" enter your e-mail, to "Password" and "Confirm password" enter password you wish to have. Your password has to contain letters big and small, at least one digit and at least one special character (e.p. !@#$%^&*). After that click Register. Your account has been created.



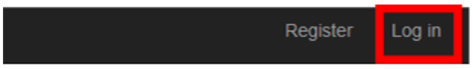
Register.

Create a new account.

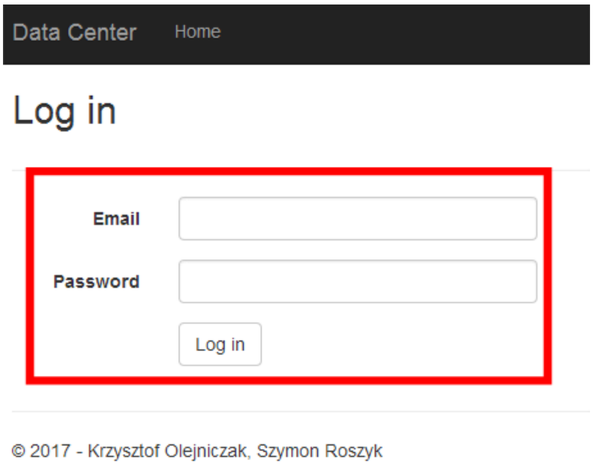
© 2017 - Krzysztof Olejniczak, Szymon Roszyk

## Log in

Firstly you have to click "Log in" button on right-top side of screen.



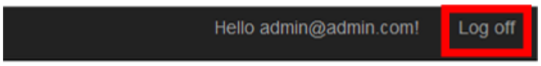
Then log in screen will apear. Now you have to fill following fields: "Email" and "Password". Of course you have to use the data given during register. Then click "Log in" button. If your account credentials are valid you will be logged in to the system.



© 2017 - Krzysztof Olejniczak, Szymon Roszyk

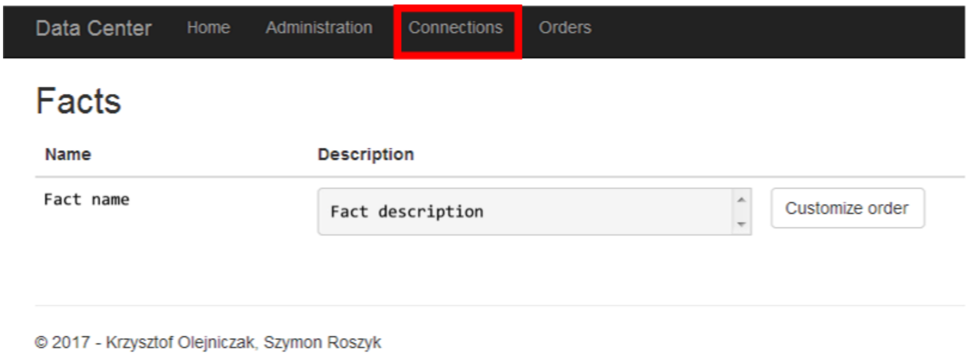
## Log off

Log off is pretty simple. All you have to do is click "Log off" on right-top side of the page.



## Creating connection and reloading database

To do that firstly you have to log in. If you already did that you have to click on "Connection" button on the top of the site.

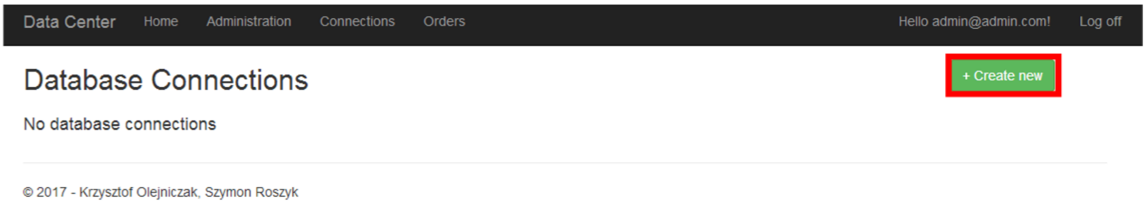


Facts

| Name | Description |
|---|---|
| Fact name | Fact description |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Next click on "Create new" button on the right.

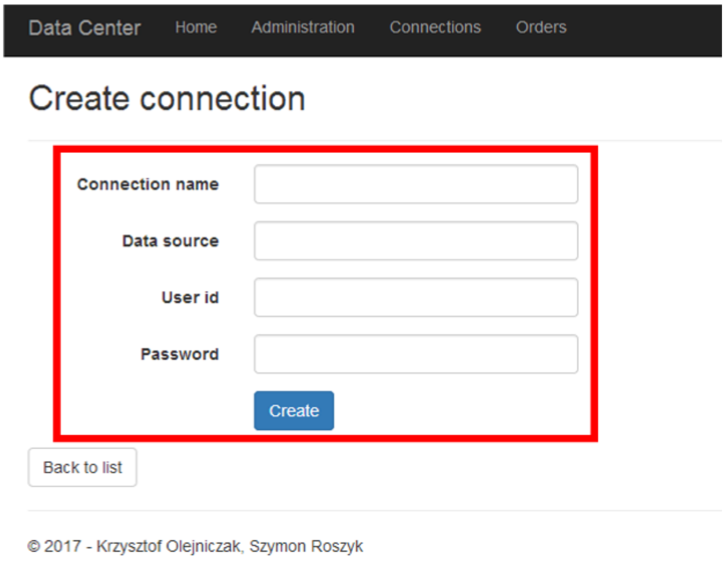


Database Connections

No database connections

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Now enter data that will make ability to enter to your database and click "Create".



Create connection

Connection name

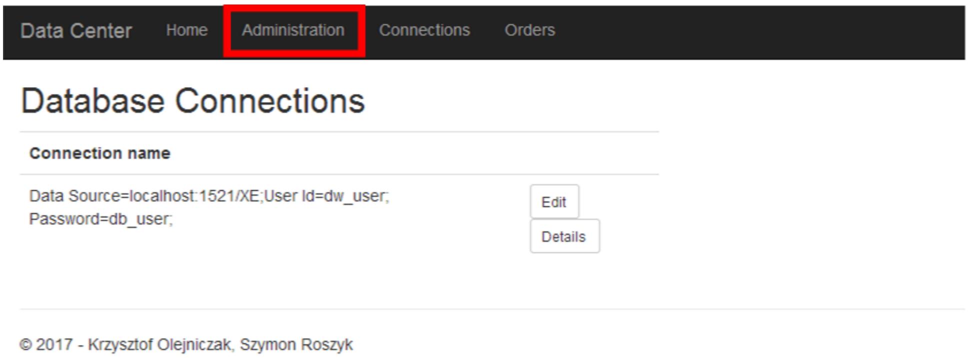
Data source

User id

Password

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

The connection has been created. You can show details of it or edit it if you wish. There is one more thing to do: reloading database. To do that you have to go to the "Administration" (button on the top).

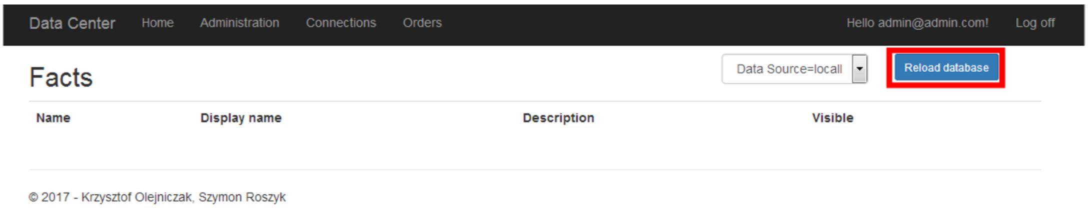


Database Connections

| Connection name |
|---|
| Data Source=localhost:1521/XE;User Id=dw_user;<br>Password=db_user; |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

After that you have to click "Reload database" button on right top of your screen. This cause getting data about tables from data warehouse. It has been described with details in chapter 5. Fact of reloaded database should be shown.



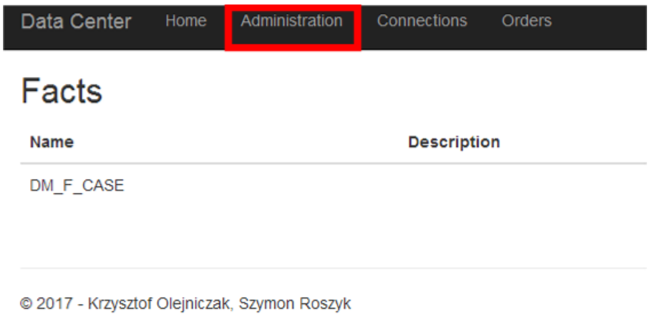
Facts

| Name | Display name | Description | Visible |
|---|---|---|---|

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

## Adding display names and description and making columns or facts or dimensions invisible to user

Now we will take care of showing to user only proper columns. To do that firstly you have to log in. Next go to "Administration" section.

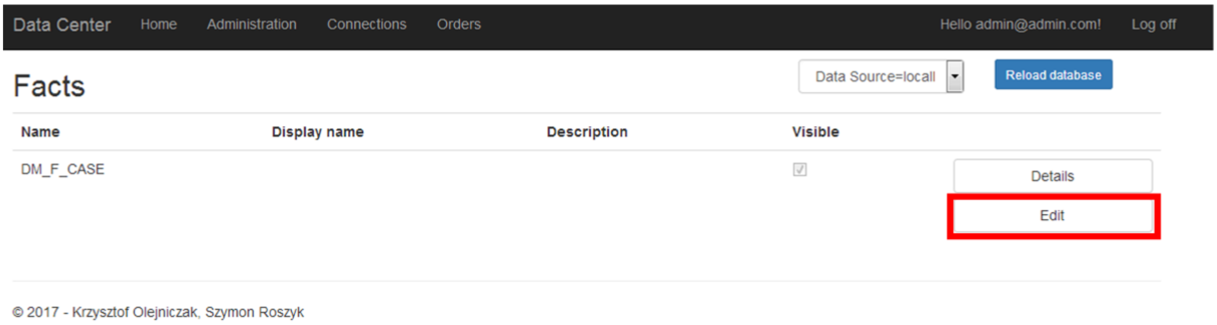


Facts

| Name | Description |
|---|---|
| DM_F_CASE | |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Next click "Edit" button near Fact that you wish to edit.

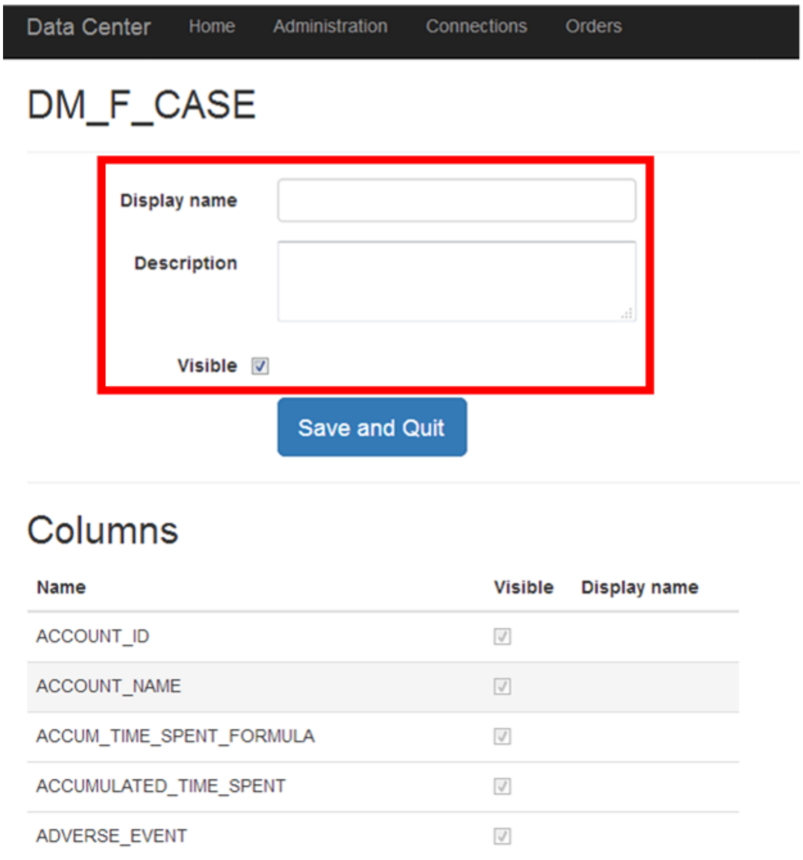


## Facts

| Name | Display name | Description | Visible |
|---|---|---|---|
| DM_F_CASE | | | ☑ |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Next you will see form that makes the ability to change properties of fact. There you can make fact invisible. You can also add description to it or edit it and you can add display name.



## DM_F_CASE

Display name

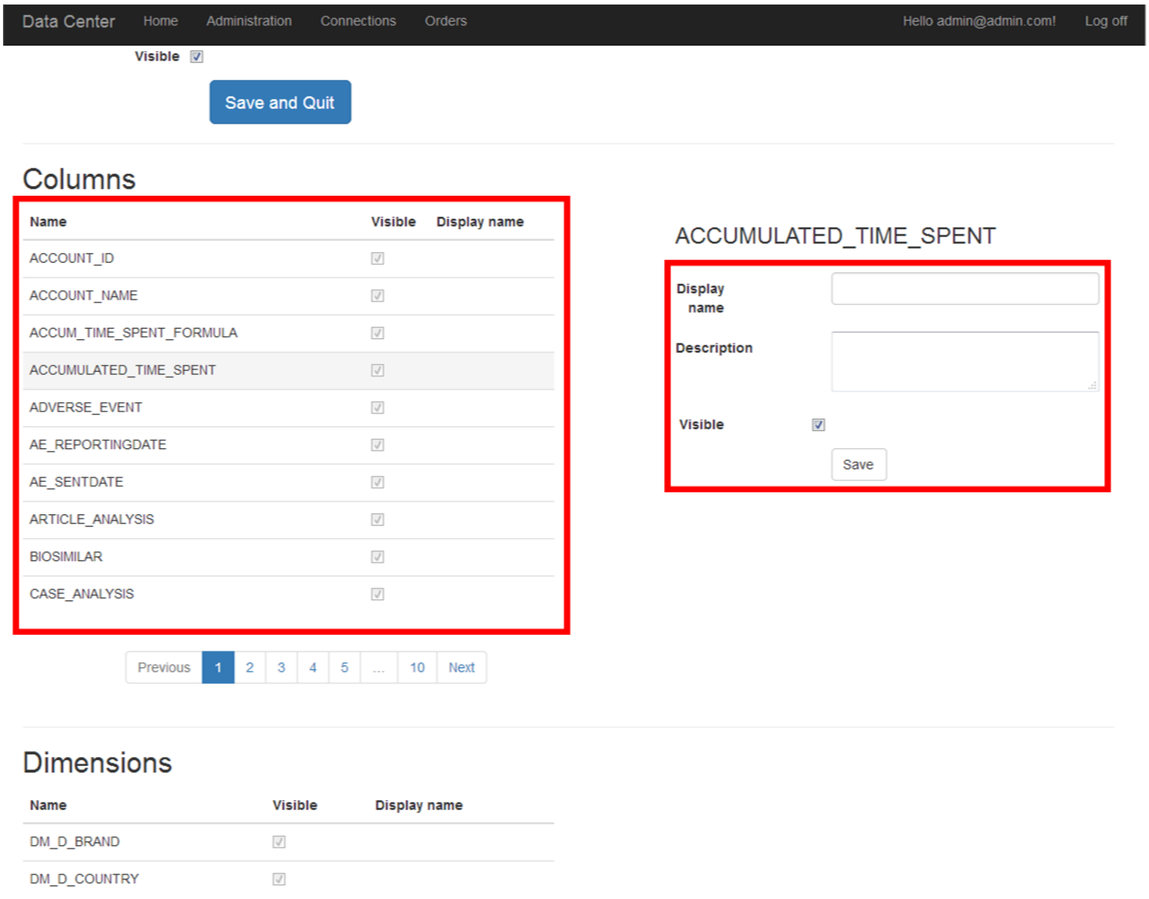
Description

Visible ☑

## Columns

| Name | Visible | Display name |
|---|---|---|
| ACCOUNT_ID | ☑ | |
| ACCOUNT_NAME | ☑ | |
| ACCUM_TIME_SPENT_FORMULA | ☑ | |
| ACCUMULATED_TIME_SPENT | ☑ | |
| ADVERSE_EVENT | ☑ | |

Below there will be shown columns of the fact. You can change their properties by clicking on them. Form on the right will appear then. There you can set up display name, change description or set column visible or invisible to user. Then you have to click "Save" button to make change.



Visible ☑

Columns

| Name | Visible | Display name |
|---|---|---|
| ACCOUNT_ID | ☑ | |
| ACCOUNT_NAME | ☑ | |
| ACCUM_TIME_SPENT_FORMULA | ☑ | |
| ACCUMULATED_TIME_SPENT | ☑ | |
| ADVERSE_EVENT | ☑ | |
| AE_REPORTINGDATE | ☑ | |
| AE_SENTDATE | ☑ | |
| ARTICLE_ANALYSIS | ☑ | |
| BIOSIMILAR | ☑ | |
| CASE_ANALYSIS | ☑ | |

ACCUMULATED_TIME_SPENT

Display name

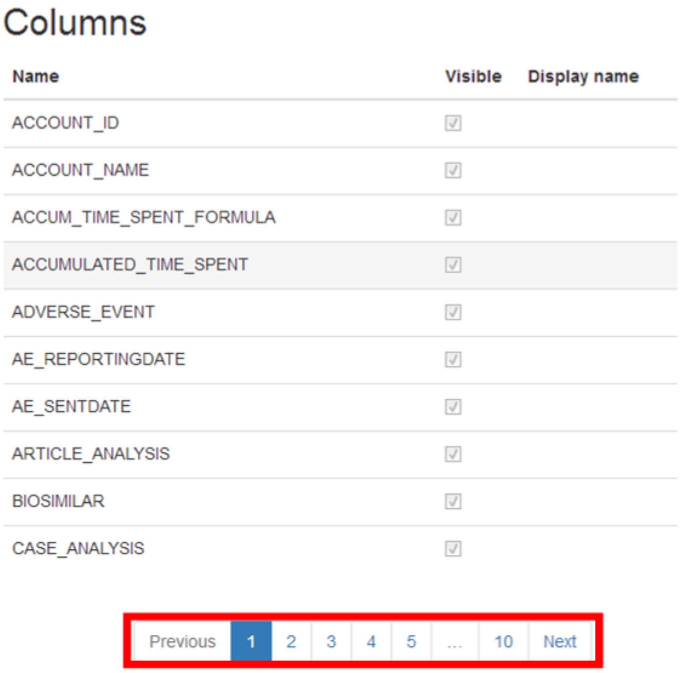
Description

Visible ☑

Dimensions

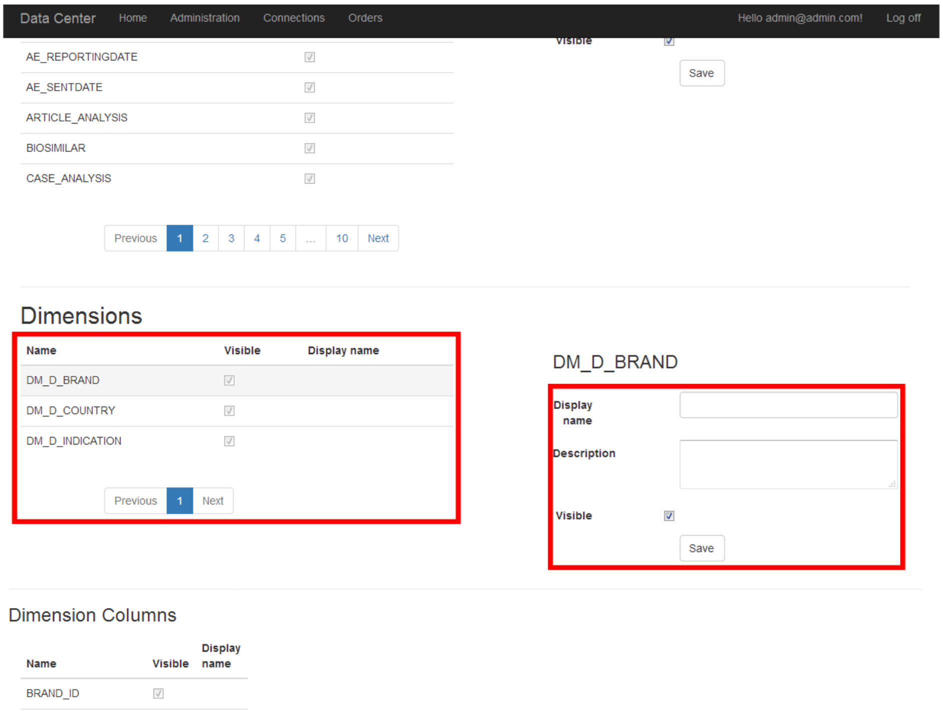
| Name | Visible | Display name |
|---|---|---|
| DM_D_BRAND | ☑ | |
| DM_D_COUNTRY | ☑ | |

Columns of the fact are divided into pages, you can switch them below list.


Columns

| Name | Visible | Display name |
|---|---|---|
| ACCOUNT_ID | ☑ | |
| ACCOUNT_NAME | ☑ | |
| ACCUM_TIME_SPENT_FORMULA | ☑ | |
| ACCUMULATED_TIME_SPENT | ☑ | |
| ADVERSE_EVENT | ☑ | |
| AE_REPORTINGDATE | ☑ | |
| AE_SENTDATE | ☑ | |
| ARTICLE_ANALYSIS | ☑ | |
| BIOSIMILAR | ☑ | |
| CASE_ANALYSIS | ☑ | |

To change dimensions you have to scrol below. There is a list of available dimensions. You can change their settings by clicing on them. Then form on the right will appear. On it you can change dispaly name, description and visibility for customer. After that click "Save" button. List of dimensions is divided into pages that you can switch below the list.



| | |
|---|---|
| AE_REPORTINGDATE | ☑ |
| AE_SENTDATE | ☑ |
| ARTICLE_ANALYSIS | ☑ |
| BIOSIMILAR | ☑ |
| CASE_ANALYSIS | ☑ |

Dimensions

| Name | Visible | Display name |
|---|---|---|
| DM_D_BRAND | ☑ | |
| DM_D_COUNTRY | ☑ | |
| DM_D_INDICATION | ☑ | |

DM_D_BRAND

Display name

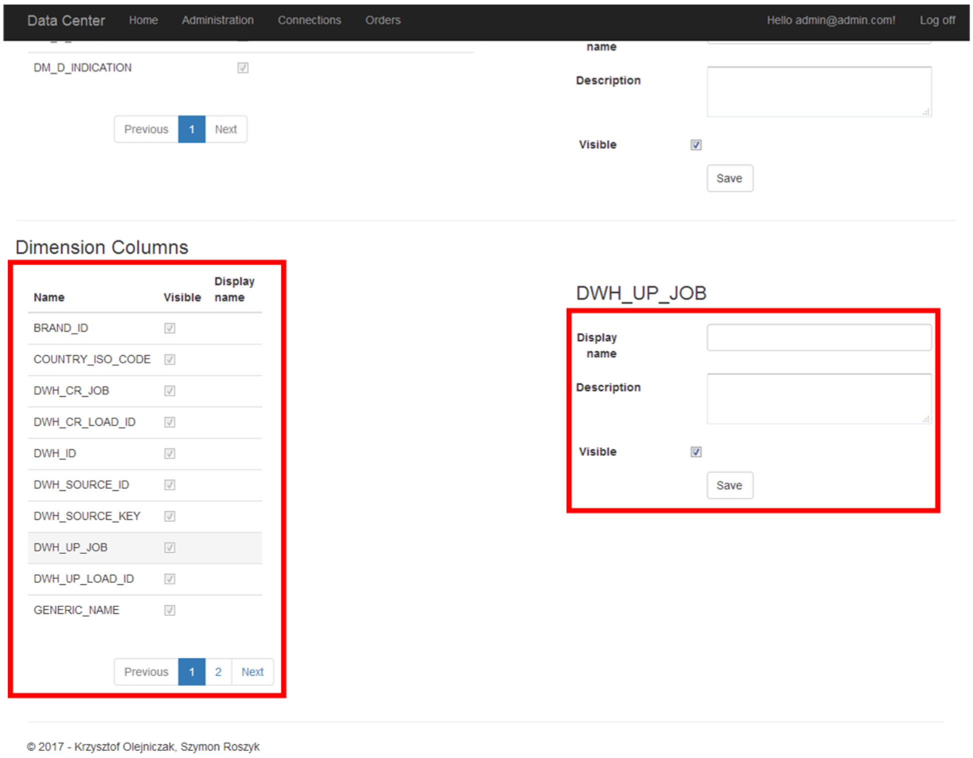
Description

Visible ☑

Dimension Columns

| Name | Visible | Display name |
|---|---|---|
| BRAND_ID | ☑ | |

After clicking on specified dimension the list of dimension's columns also appears. You can easly edit them by clicing on them and do the same things that you did during editing columns of the fact.



| | |
|---|---|
| DM_D_INDICATION | ☑ |

Dimension Columns

| Name | Visible | Display name |
|---|---|---|
| BRAND_ID | ☑ | |
| COUNTRY_ISO_CODE | ☑ | |
| DWH_CR_JOB | ☑ | |
| DWH_CR_LOAD_ID | ☑ | |
| DWH_ID | ☑ | |
| DWH_SOURCE_ID | ☑ | |
| DWH_SOURCE_KEY | ☑ | |
| DWH_UP_JOB | ☑ | |
| DWH_UP_LOAD_ID | ☑ | |
| GENERIC_NAME | ☑ | |

DWH_UP_JOB

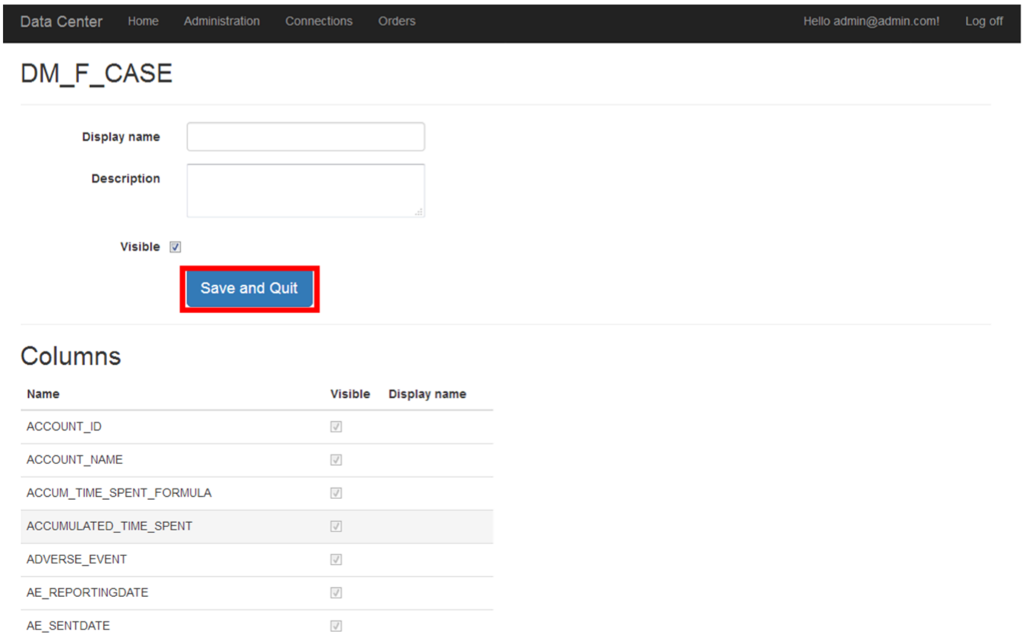
Display name

Description

Visible ☑

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

You have to click "Save and Quit" button after all if you wish changes to take effect. This button is below fact details modification form.



DM_F_CASE

Display name

Description

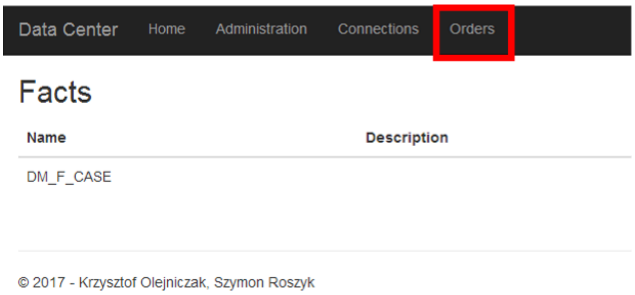
Visible ☑

Columns

| Name | Visible | Display name |
|---|---|---|
| ACCOUNT_ID | ☑ | |
| ACCOUNT_NAME | ☑ | |
| ACCUM_TIME_SPENT_FORMULA | ☑ | |
| ACCUMULATED_TIME_SPENT | ☑ | |
| ADVERSE_EVENT | ☑ | |
| AE_REPORTINGDATE | ☑ | |
| AE_SENTDATE | ☑ | |

## Viewing orders and changing their status

To do that you have to be logged in. Then click on the "Orders" button on the top.

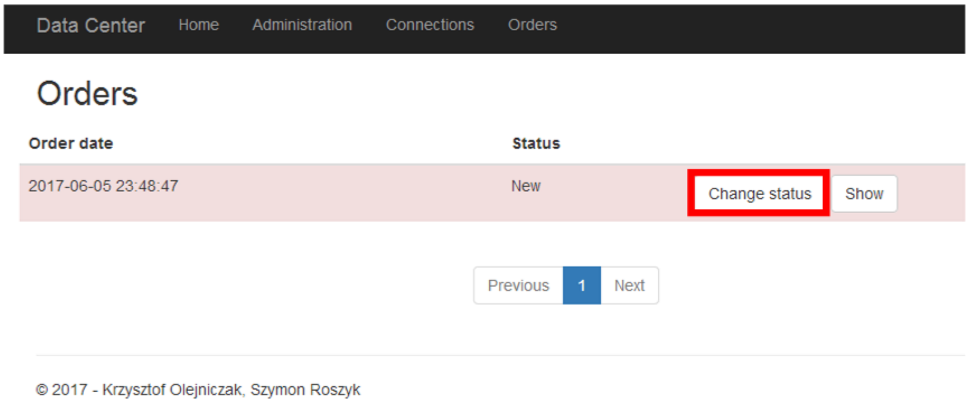


Facts

| Name | Description |
|---|---|
| DM_F_CASE | |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Next, list of the orders will be shown. Orders that are new will be red, orders in progress will be yellow and orders that are completed will be green. To change status of specified order click on "Change status" button.

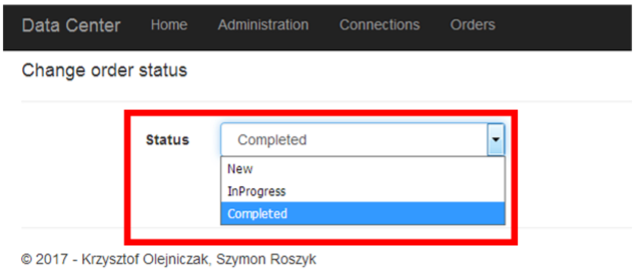


Orders

| Order date | Status | |
|---|---|---|
| 2017-06-05 23:48:47 | New | Change status Show |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Then you can choose new status of order using drop-down list. To complete this click on "Save" button.

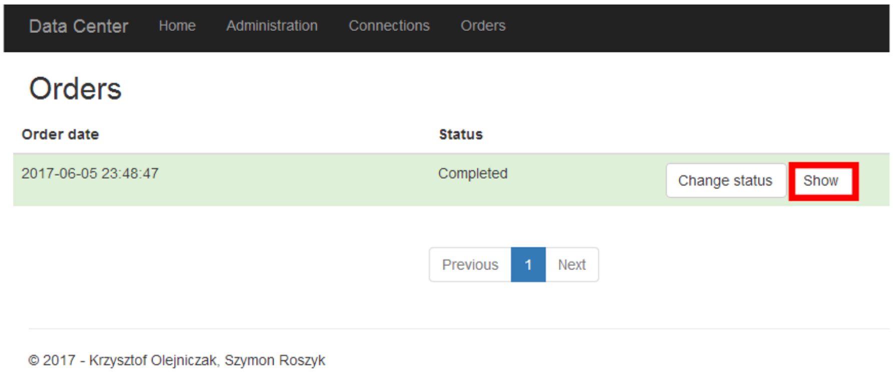


Change order status

Status: Completed

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

You returned to list of orders. To show what exactly customer ordered click on "Show" button beside specified order.

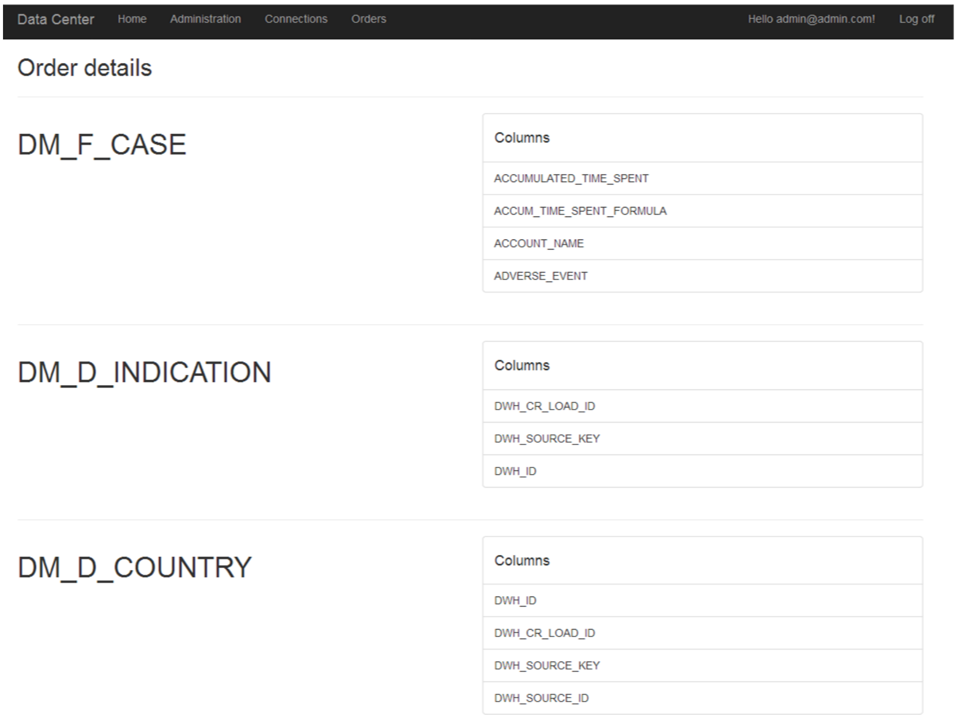


Orders

| Order date | Status | |
|---|---|---|
| 2017-06-05 23:48:47 | Completed | Change status Show |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

There you can see details of customer's order with fact's columns, it's dimensions and columns of those dimensions.



Order details

DM_F_CASE

| Columns |
|---|
| ACCUMULATED_TIME_SPENT |
| ACCUM_TIME_SPENT_FORMULA |
| ACCOUNT_NAME |
| ADVERSE_EVENT |

DM_D_INDICATION

| Columns |
|---|
| DWH_CR_LOAD_ID |
| DWH_SOURCE_KEY |
| DWH_ID |

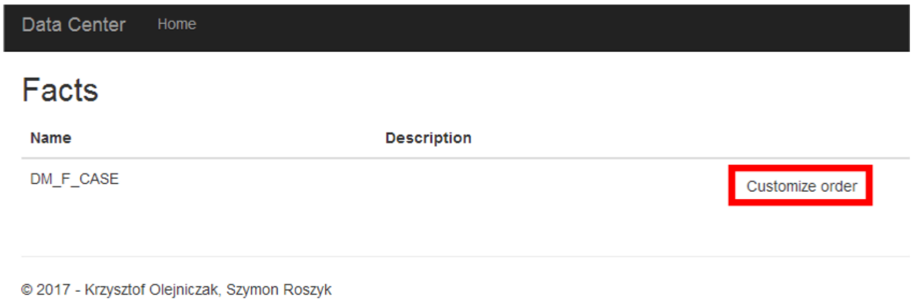
DM_D_COUNTRY

| Columns |
|---|
| DWH_ID |
| DWH_CR_LOAD_ID |
| DWH_SOURCE_KEY |
| DWH_SOURCE_ID |

### 8.2.2 Customer's tools

### Creating order

Our customer need to make order. This is the most important part of our system. The list of facts is on home page. To make order click "Customize order" button beside fact that you wish to order.



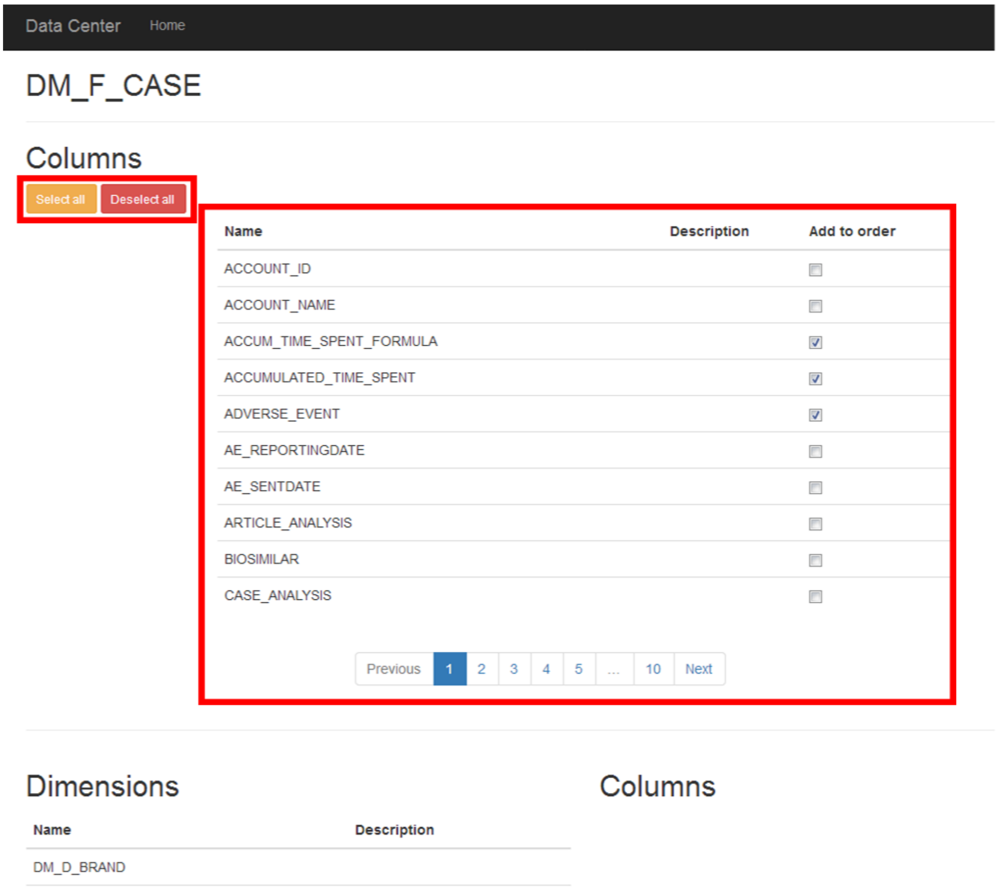
Data Center Home

Facts

| Name | Description | |
|---|---|---|
| DM_F_CASE | | Customize order |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Then you can see the list of columns of the fact. You can add to order specified columns by selecting checkbox on "Add to order" column. If you want you can select all columns at once using "Select all" button on the top. You can also deselect all columns using "Deselect all". List of columns is divided into pages that you can swith below.



Data Center Home

DM_F_CASE

Columns

Select all Deselect all

| Name | Description | Add to order |
|---|---|---|
| ACCOUNT_ID | | ☐ |
| ACCOUNT_NAME | | ☐ |
| ACCUM_TIME_SPENT_FORMULA | | ☑ |
| ACCUMULATED_TIME_SPENT | | ☑ |
| ADVERSE_EVENT | | ☑ |
| AE_REPORTINGDATE | | ☐ |
| AE_SENTDATE | | ☐ |
| ARTICLE_ANALYSIS | | ☐ |
| BIOSIMILAR | | ☐ |
| CASE_ANALYSIS | | ☐ |

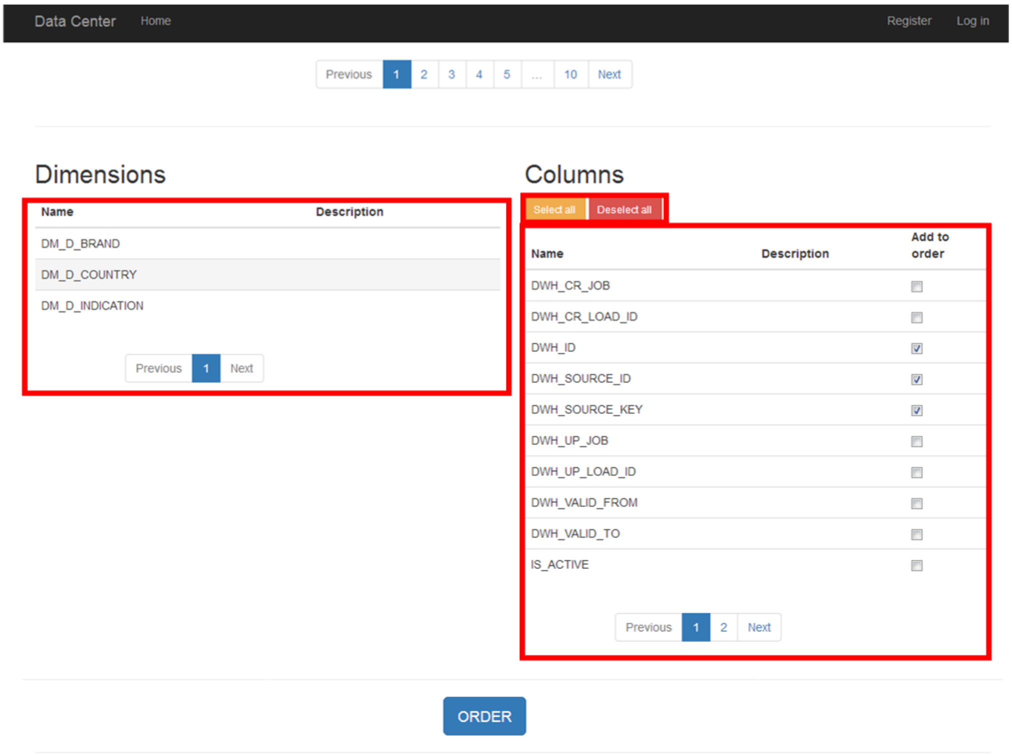
Previous 1 2 3 4 5 ... 10 Next

Dimensions

| Name | Description |
|---|---|
| DM_D_BRAND | |

Columns

On the bottom there is list of dimensions. You can show and switch list of columns of specified dimension by clicking on dimension list. The list of columns is also divided into pages. You can select columns from list if you wish to add them into order. If you want you can select all columns at once using "Select all" button on the top. You can also deselect all columns using "Deselect all".



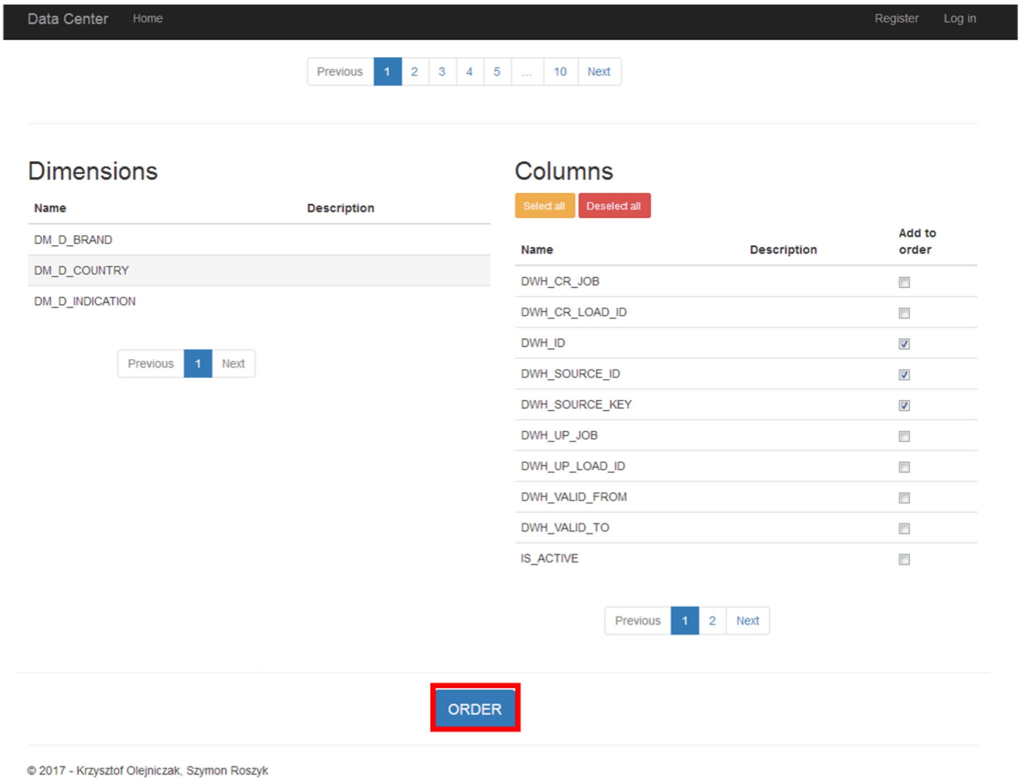
Dimensions

| Name | Description |
|---|---|
| DM_D_BRAND | |
| DM_D_COUNTRY | |
| DM_D_INDICATION | |

Columns

| Name | Description | Add to order |
|---|---|---|
| DWH_CR_JOB | | □ |
| DWH_CR_LOAD_ID | | □ |
| DWH_ID | | ☑ |
| DWH_SOURCE_ID | | ☑ |
| DWH_SOURCE_KEY | | ☑ |
| DWH_UP_JOB | | □ |
| DWH_UP_LOAD_ID | | □ |
| DWH_VALID_FROM | | □ |
| DWH_VALID_TO | | □ |
| IS_ACTIVE | | □ |

To realize order click "ORDER" button below.



Dimensions

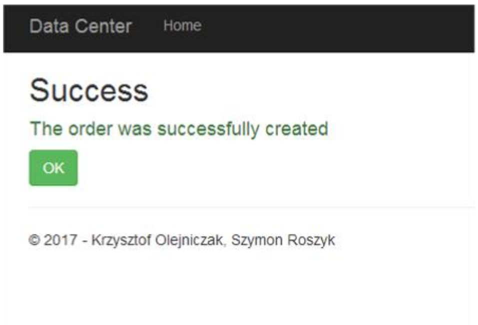
| Name | Description |
|---|---|
| DM_D_BRAND | |
| DM_D_COUNTRY | |
| DM_D_INDICATION | |

Columns

| Name | Description | Add to order |
|---|---|---|
| DWH_CR_JOB | | □ |
| DWH_CR_LOAD_ID | | □ |
| DWH_ID | | ☑ |
| DWH_SOURCE_ID | | ☑ |
| DWH_SOURCE_KEY | | ☑ |
| DWH_UP_JOB | | □ |
| DWH_UP_LOAD_ID | | □ |
| DWH_VALID_FROM | | □ |
| DWH_VALID_TO | | □ |
| IS_ACTIVE | | □ |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

If there will be success message on screen that mean that your order is made sucessfully.



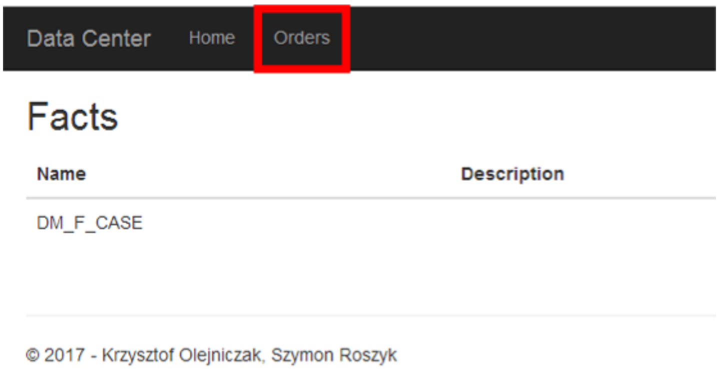
Success

The order was successfully created

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

## Viewing orders and their details

To show past orders click on the "Orders" button on the top.

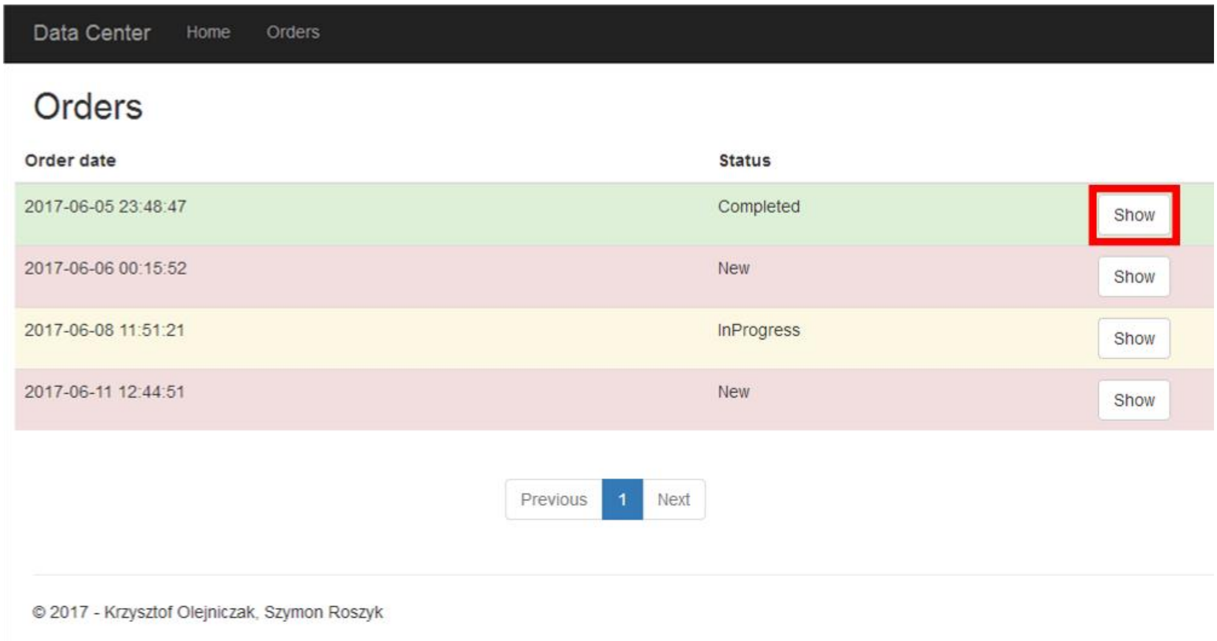


Facts

| Name | Description |
|---|---|
| DM_F_CASE | |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

Next, list of the orders will be shown. Orders that are new will be red, orders in progress will be yellow and orders that are completed will be green. To show what exactly was ordered click on "Show" button beside specified order.

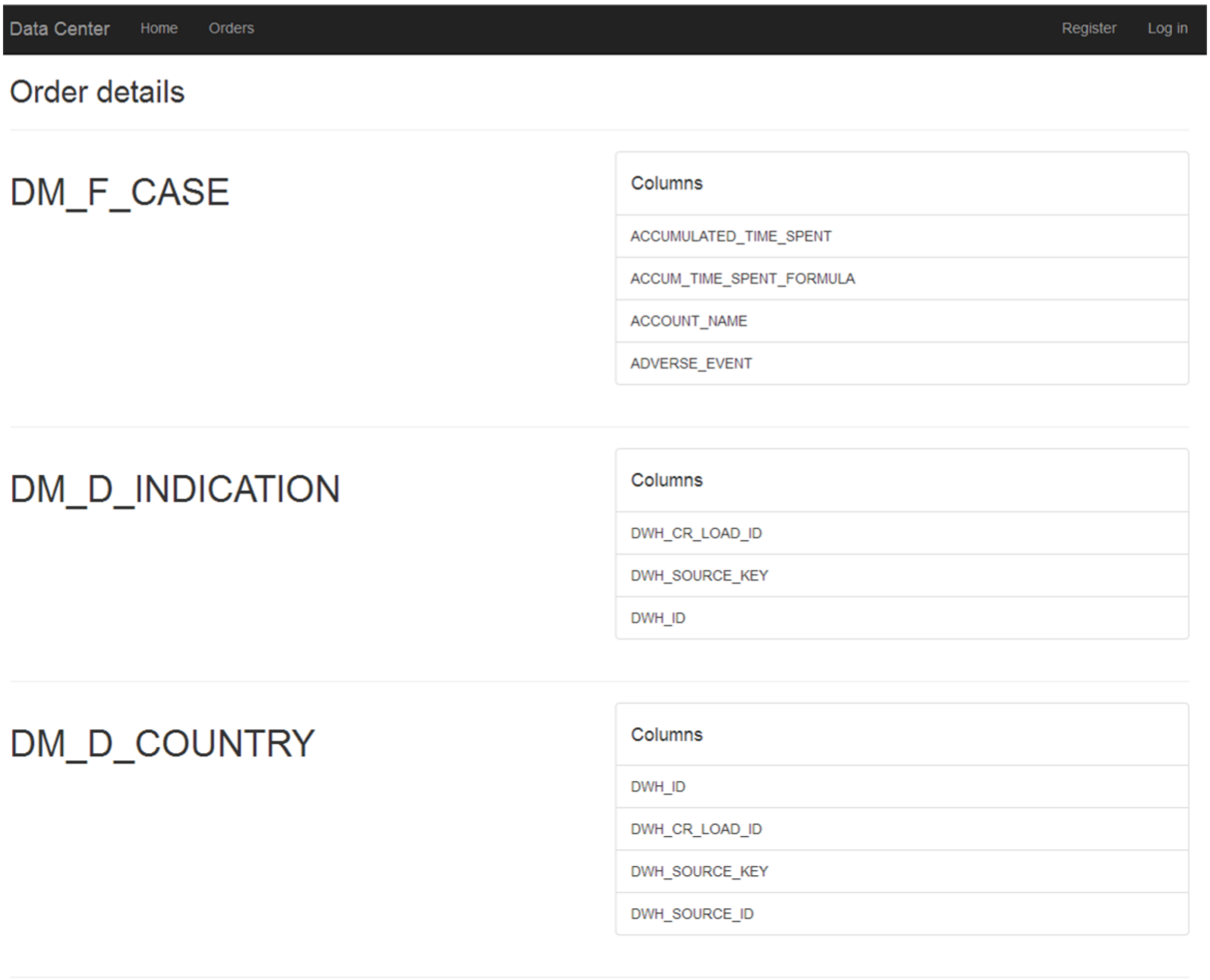


Orders

| Order date | Status | |
|---|---|---|
| 2017-06-05 23:48:47 | Completed | Show |
| 2017-06-06 00:15:52 | New | Show |
| 2017-06-08 11:51:21 | InProgress | Show |
| 2017-06-11 12:44:51 | New | Show |

© 2017 - Krzysztof Olejniczak, Szymon Roszyk

There you can see details of customer's order with fact's columns, it's dimensions and columns of those dimensions.



Order details

DM_F_CASE

| Columns |
|---|
| ACCUMULATED_TIME_SPENT |
| ACCUM_TIME_SPENT_FORMULA |
| ACCOUNT_NAME |
| ADVERSE_EVENT |

DM_D_INDICATION

| Columns |
|---|
| DWH_CR_LOAD_ID |
| DWH_SOURCE_KEY |
| DWH_ID |

DM_D_COUNTRY

| Columns |
|---|
| DWH_ID |
| DWH_CR_LOAD_ID |
| DWH_SOURCE_KEY |
| DWH_SOURCE_ID |

# 9 Development prospects

Making of our application is open. There is many things that could make program better and with more functionality.

## Communication with data warehouses using different drivers

Today our application operate only on Oracle Database. It would be much better if it operate also on others databases like DB2, Microsoft SQL Server and others database systems.

## Better interface for Administrator that enables realizing orders

We could also allow Administrator realizing order via application sending data to client. He could see data in tables, customize them if he wish and send to customer.

## Recommendation system

The system would recommend to user what data would be useful for him. It could be shown on main page or in fact view page and it could base on what customer is that.

## Generate part result from complex order.

Big order is huge work for Administrator to do. Doing it manually would be very annoying. The idea is to generate part of result from complex orders by system. It would save much time of work.