

Tryout Tutorial

Contents

1	Intr	oducti	on to i-views	4
	1.1	The K	nowledge Builder	4
	1.2	Install	ation and launching the Knowledge-Builder	5
		1.2.1	Launching the Knowledge-Builder in Windows	5
		1.2.2	Launching the Knowledge-Builder in Mac	7
		1.2.3	Launching the Knowledge-Builder in Linux	9
	1.3	Explar	nation: The Knowledge Graph in the Knowledge-Builder	11
2	The	eleme	ents of the Knowledge Graph	13
	2.1	Creati	ng and displaying your own objects in the Knowledge Graph	13
		2.1.1	Creating a new object (type "organization")	13
	2.2	Editin	g the newly created company	13
		2.2.1	Assigning a new attribute	13
		2.2.2	Searching for the newly created organisation	14
	2.3	Addin	g a relation	15
	2.4	End o	f chapter and outlook	16
3	The	graph	-editor	17
	3.1	Displa	ying the Knowledge Graph	17
		3.1.1	Displaying objects, types and relations with the graph-editor \ldots .	17
		3.1.2	Displaying objects and object types via drag & drop	19
	3.2	Drawi	ng and deleting relations in the graph-editor	20
		3.2.1	Drawing relation	20
		3.2.2	Deleting a relation	23
	3.3	End o	f chapter and outlook	24
4	Def	inition	of the Knowledge Graph schema	24
	4.1	Creati	ng and editing new types	25
	4.2	Defini	ng and using attributes	26
		4.2.1	Defining a new attribute schema	26
		4.2.2	Assigning the new attribute	28
	4.3	Defini	ng and establishing relations	28
		4.3.1	Defining a new relation schema	28
		4.3.2	Assigning the new relation	30
	4.4	End o	f chapter and outlook	31



5	Sea	rching in the Knowledge Graph	32
	5.1	Searching objects (Tip)	32
	5.2	Creating and displaying a search folder	32
	5.3	Structured queries	32
		5.3.1 Creating a simple structured search	32
		5.3.2 Creating a complex structured search	34
		5.3.3 Creating a new structured query	35
	5.4	End of chapter and outlook	38
6	Buil	ding your own models	39
	6.1	Import and Export of data from external sources	39
		6.1.1 Example: Import of persons from a CSV file	39
	6.2	Other Knowledge Graphs included in this tryout version	40
	6.3	Any questions or feedback?	41
	6.4	Contact	41

1 Introduction to i-views

Welcome to our tutorial

We are glad that you have downloaded the i-views tryout version.

By means of the i-views Knowledge-Builder, you experience a user friendly software for administering data within a Knowledge Graph platform. The advantage of a semantic knowledge graph database is the multidimensional linking of data as well as the ability to explore data interrelationships graphically.

With this tutorial we will guide you through i-views, step by step. You will get to know the basics as well as features of our software.

This tutorial is addressed to system administrators as well as to data modelers who whish to get a fast overview about the functionality of i-views.

Please take approx. 40 minutes of your time to process all **six sections** of the tutorial. You will see that you don t need any programming skills to use the graph database.

Please do not hesitate to contact us in case you have any questions: support@empolis.com

1.1 The Knowledge Builder

This tutorial introduces a Knowledge Graph, produced by the company intelligent views. This example enables you to quickly understand how a Knowledge Graph works and the possibilities offered by i-views for setting it up and maintaining it.

The tutorial is addressed to system administrators and data modelers who want to get a quick overview about the functionality of i-views.

TIP:This tryout-version consists of modelling environment and server. For further information about the architecture of i-views, APIs, REST and other technical topics, please consult the i-views-documentation.

Tutorial-Specials



Throughout the tutorial you will come across this illustration. With every section you have finished one circle will fill up - thus you can keep track on your training status and success at all times.



So let s start. You will get to know the following contents:

- Installation and launching of the Knowledge-Builder
- The Knowledge Graphin the Knowledge-Builder

We wish you lots of fun!

1.2 Installation and launching the Knowledge-Builder

1.2.1 Launching the Knowledge-Builder in Windows

1. Download the i-views tryout version zip file

i-views

2. Unpack the i-views tryout zip file with a click on the menu item *extract all* of the context menu. Open the context menu by making a right click on the zip file.

	i-views
-	i-views

3. Double-click on the directory "i-views_win" to find the following five objects:

	resources
	volumes
8	i-views-tryout
	i-views-tryout
2	i-views-tryout_en

4. Launch the **i-views-tryout.exe** application per double-click and the start dialogue will be displayed:

	resources
	🔒 volumes
<	i-views-tryout
	i-views-tryout
	🔁 i-views-tryout_en

5. Click besides the input field of *Knowledge Graph* and select the Knowledge Graph **i-views-tryout**. Enter *admin* in the *User* field. For this tryout-version, leave the *Password* field blank. Click the button *Start* to confirm your data entries.



	First time here?		
	Getting started		
Server	localhost:64189		
Semantic network			
User			
Password			
REST Port	8815		
Start Quit		<u>A</u> bout	

NOTE:Since the tryout version functions as a server, you might get a security warning as well as a windows firewall warning, which you have to confirm in order to run i-views.

🔐 Windows-Sicherheitshinweis 🛛 📔					
Die Windows-Firewall hat einige Funktionen dieses Programms blockiert.					
Einige Features vor Netzwerken und Do	n i-views-tryout w mänennetzwerke	vurden in allen öffentlichen Netzwerken, privaten en von der Windows-Firewall blockiert.			
8	Name:	i-views-tryout			
000	Herausgeber:	intelligent views gmbh			
	Pfad:	C: \users\public\j-views\j-views-tryout.exe			
Kommunikation von	Kommunikation von i-views-tryout in diesen Netzwerken zulassen:				
Private Netz	werke, beispielsw	eise Heim- oder Arbeitsplatznetzwerk			
Öffentliche Netzwerke, z. B. in Flughäfen und Cafés (nicht empfohlen, da diese Netzwerke oftmals gar nicht oder nur geringfügig geschützt sind)					
Welche Risiken bestehen beim Zulassen eines Programms durch eine Firewall?					
🚱 Zugriff zulassen 🛛 Abbrechen					

6. Click the button Allow access and the main window of i-views will open.



TIP:You will find further information for Knowledge-Builder settings, especially for user administration, consulting the i-views-documentation.

1.2.2 Launching the Knowledge-Builder in Mac

1. Download the i-views tryout version zip file.

i-views_mac_os.zip

2. Unpack the i-views tryout version zip file by double-clicking on the file.

i-views_mac_os
i-views_mac_os.zip

3. Double-click to go to the directory "i-views_mac_os".



4. Open your user folder by **Cmd + Shift + H**. Copy the directory "i-views" into it, using the shortcut key **Cmd + C** and, after changing to the user folder, **Cmd + V**.

►	🛅 Desktop	
►	Documents	
►	💽 Downloads	
►	i-views	
►	Movies	
►	Music	
►	Pictures	
►	Public	

.

5. Launch the application i-views-tryout. Click besides the input fields of *Knowledge Graph* and select the Knowledge Graph *i-views-tryout*. Enter *admin* in the *User* field. For this tryout-version, leave the *Password* field blank. Click the button *Start* to confirm your data entries.

8	First time here?		
	Getting started		
Server	localhost:64189		
Semantic network			
User			
Password			
REST Port	8815		
Start Quit	A	bo	out

NOTE: When starting i-views for the first time, a window appears in which the user is asked whether he or she trusts the manufacturer. This window might be overlapped by another window.

6. Click the button *Allow access* and the main window of i-views will open.



TIP: You will find further information for Knowledge-Builder settings, especially for user administration, consulting the i-views-documentation.

1.2.3 Launching the Knowledge-Builder in Linux

1. Download the i-views tryout version zip file.

2. Open the console by pressing **Strg + Shift + T**.

3. Go to the directory which contains the downloaded i-views tryout version. Use the following command:

cd / .../download_directory

...

4. Unpack the i-views tryout version zip file using the following command:

unzip i-views_linux.zip

5. Afterwards go to the newly created directory "i-views_linux" using the following command: **cd i-views_linux**

The directory "i-views_linux" contains the following files:

i-views-tryout_en.pdf	i-views-tryout.pdf	resources
i-views-tryout.ini	i-views-tryout.sh	volumes

To display the content of the directory use the following command:

ls

6. Launch the file "i-views-tryout.sh" using the following command:

./i-views-tryout.sh

If the file doesn t launch, the reasons may be insufficient access rights. Type in the following command and try launching the file again:

chmod 777 i-views-tryout.sh

7. Click besides the input fields of *Knowledge Graph* and select the Knowledge Graph *i-views-tryout*. Enter *admin* in the *User* field. For this tryout-version, leave the *Password* field blank. Click the button *Start to* confirm your data entries.

8	First time here?		
	Getting started		
Server	localhost:64189		
Semantic network			
User			
Password			
REST Port	8815		
Start Quit	Ab	out	

8. Click the button *Allow access* and the main window of i-views will open.



...

TIP: You will find further information for Knowledge-Builder settings, especially for user administration, consulting the i-views-documentation.

1.3 Explanation: The Knowledge Graph in the Knowledge-Builder

The base of a Knowledge Graph is formed by its types (person, company, location,...) and objects (**Jana Schmidt**, **Miller plc**, **Frankfurt**...), which have attributes (call number, date of birth,...) assigned to them and which are connected by relations. In the following image, the relations have been visualized:



The knowledge network in the Knowledge-Builder



The **tryout Knowledge Graph** already contains some types, objects, attributes and relations. They will help you to understand the essential functions using the Knowledge-Builder.

Congratulations! You have finished the first section successfully.



Let s move on. You will get to know the following contents:

- Creating a new object of type organization
- Editing a company
- Searching for an organization

We wish you lots of fun!



2 The elements of the Knowledge Graph

2.1 Creating and displaying your own objects in the Knowledge Graph

2.1.1 Creating a new object (type "organization")

In the i-views Knowledge Graph, all elements are organized either as a **Type** or as an **Object**. Types provide the structure for assorting all objects. At the same time, the **principle of inheritance** comes into effect: Properties which have been defined for a type will also be effective for all subordinate types (= subtypes) and for all objects that are assorted within.

HINT: In general, the structure tree on the left side shows only types. On the right side, either subtypes or objects will be shown, depending on the chosen tab.

1. Click on semantic *Knowledge Graph* in the folder panel on the left of the organizer to open the folder. All types are displayed.

2. Expand the type *Organisation* and click on the subtype *Company*.

3. On the right window, click onto *Objects* tab: You will see a list of companies that already exist. These companies are objects of the type *Company*.

4. Enter the name of the new company **Miller plc** in the input line.



5. Now click the button **•**.

Done. The company **Miller plc** is created as an object in the Knowledge Graph.

2.2 Editing the newly created company

2.2.1 Assigning a new attribute

The first task for new objects such as *companies*, *persons* or *products* is to complete the attributes assigned to them.

- 1. Select the new object *Miller plc* from the list of objects of companies. The detailed information for *Miller plc* is now displayed in the lower section of the window.
- 2. Click the button Add attribute and choose *also known as (Synonym)* from the list of available attributes.

Miller plc		Company	l⊞ Choose - □ × Choose property
Attributes Name Relations	■ Miller plc Add attribute	^	Also Known As (Synonym) Icon RDF-ID RDF-URI Telephone Number
Relations	Add relation	~	 Key Shift for multi-selection All None OK Cancel

3. Enter *Milpc* as the value for the attribute *Also known* as.

0 (

Subtypes Objects		≡*□
🕑 🖋 ⊱ 🔎 .	A X © № ™	
Miller plc Name		
Miller plc		
<		>
Miller plc	Compan	
Attributes		^
Also Known As (Synonym)	≡ Milpq	
▶ Name	■ Miller plc	
	Add attribute	
Relations		

You can add other attributes of your choice such as *Telephone Number* as described in steps 1-3.

2.2.2 Searching for the newly created organisation

As a test, search for a newly created organisation: enter the first four letters milp of the synonym in the search box, click on the magnifier symbol to the right of the search box and choose *Search Name & Synonym*.

Alternatively, you can also press RETURN instead of clicking the magnifier symbol. In this case, the search will be performed using the search option which was chosen the last time.

Milp	🖉 🏃 🗄 🖷 🖏				≡⇔□
FOLDER	Name	Туре	Reason	Search string	Quality ^
SEMANTIC KNOWLEDGE NETV	Miller plc	Company	Also Known As (Synonym)	Milp*	100
 Object Types 					
4 🥥 Location					
City					
Country					~



When searching for *milp*, the system displays *Miller plc* in the hit list, indicating that this match was found using the *Also known as*-attribute.

2.3 Adding a relation

In Knowledge Graphs relations as connections between objects are an essential feature. Use the button *Add relation* to create relations to connect objects: the relation *organisation employs person* shall be established to connect the objects *Hans Henry* and *Miller plc*.

1. In the type editor, click the button Add relation and choose the desired relation *organisation employs person* from the list by clicking:

🗱 Choose			_		×
Choose proper	ty				
			N	()	A
organisatio	on employs	persor	(Instance	es of P	erso
organisatio	on is located	d in (In	stances of	f Locat	tion
organisatio	on is part of	fsecto	r (Instance	es of S	ecto
Start sema	intic elemer	nt of (li	nstances o	of Flexi	ble
<					>
Hold down Str	g/Shift for multi-	selection			
Show details					
All	None		ОК	Car	ncel

2. Enter the name *Hans Henry* in the input line next to the relation and confirm with RERTURN.

Milp	🖉 🏃 🖩 🗄 🖏						≣≉⊡
FOLDER	Name	Туре	5	Reason	Search string	Quality	^
SEMANTIC KNOWLEDGE	Miller plc	Com	npany	Also Known As (Synonym)	Milpc	100	
 Object Types 							
Location							
🕨 💜 Object							
🖌 🕮 Organisation							
Company							
Institution	<						>
Person							
🦘 Project	willer pic						
🕨 💡 Topic	Attributos						^
Relation types	Attributes						
Attribute Types	Also Known As (Synonyr	n) ≡	Milpc				
TECHNICAL	Name	≡	Miller plc				
🕨 📾 Rights			Add attribute				
[★] Trigger ~			Add attribute				
< >>	Relations						
Community	organisation employs per	son ≡	Hans Henry				2
~			Add relation				~

3. If the object (here *Hans Henry*) does not yet exist, it can be created directly as a relation target. In this case a dialogue has to be confirmed with OK.

Create Hans Henry as target?						
Yes	No					

The new object *Hans Henry* as an employee of *Miller plc* is now part of the Knowledge Graph.

2.4 End of chapter and outlook

Congratulations! You have finished the second section successfully.



Let s move on. You will get to know the following contents:

- Displaying objects, types and relations with the graph-editor
- Drawing and deleting relations with the graph-editor



We wish you lots of fun!

3 The graph-editor

3.1 Displaying the Knowledge Graph

The Knowledge-Builder is a tool for Knowledge Graphs, but so far we have only talked in terms of lists and folders.

The Knowledge-Builder provides the **graph-editor** for graphical navigable views of the Knowledge Graph. It not only offers a view but it is also an effective tool for editing the Knowledge Graph, i.e. creating and deleting types, establishing relations and much more.

3.1.1 Displaying objects, types and relations with the graph-editor

- 1. Select the object *Miller plc* in the search results list.
- 2. Click the button 🕟.

Milp	8 > 1 = 5		
FOLDER	Name	Туре	Reason
SEMANTIC KNOWLEDGE	Miller plc	Company	Also Known As (Synonym)
 Object Types Location Object Organisation Company 			
Institution	<		

The *graph-editor* is opened, displaying the object **Miller plc**.

3. Position the mouse cursor over **Miller plc**. Two small — icons will appear - one at the top and one to the left of the object.



Graph View Selection					
★☆×	≻≌<>		max. Nodes	5	•
Working Folder (
Legend					
Company		+ Miller ptc			
Overview					
٠					
	re-use last relation				

4. Double-click on the left-hand plus-icon. The relation between Miller plc and Hans Henry will be displayed: **Hans Henry** is an employee of the company **Miller plc**.

Graph View Selection				
★★×	≻ 	max. Nodes 5	•	
Working Folder (Private				
Legend	Millor plc			
^	Willier pic			
Person				
Company				
Company	٩			
¥				
Overview		Hans Henry		
	re-use last relation			
<			>	



All relations to other objects will be shown by clicking at the \square -icon. If all relations are already displayed the \square -icon is used instead of the plus-icon.

3.1.2 Displaying objects and object types via drag & drop

- 1. Place the organizer window next to the graph-editor window. You can now drag & drop types and objects from the folders or from the search results box into the graph-editor.
- 2. Enter the type **London** in the search window and perform a simple search by pressing RETURN.

London	8 > 🗄 🗄 🖏				≡≉⊡
FOLDER	Name	Туре	Reason	Search string	Quality
SEMANTIC KNOWLEDGE	London	City	Name (GER)	London	100
▲ ○ Object Types					
Location					
🕨 🎔 Object					
Interpretended in the second secon					
Company					
Institution	<				>
Person	London				City
Sect → Project	LONGON				\sim
Topic	Attributes				^
 Relation types Attribute Trace 	Namo	= London			
$\bullet \bigtriangleup$ Attribute Types	* Name				
TECHNICAL		Add attribute			
🕨 📾 Rights	Relations				
[™] Trigger v					
Community		Add relation			
^					
~					~

3. In the search results, highlight **London** and drag it, by holding down the left mouse button, into the graph-editor window, then drop it.





By doing so, you can place more objects from the organizer in the graph-editor window and thus examine a greater number of objects in their semantic context.

3.2 Drawing and deleting relations in the graph-editor

Relations represent connections between different objects and are therefore a core element of the Knowledge Graph. The current graph-editor window contains the objects **Hans Henry**, **Miller plc** and **London**.

3.2.1 Drawing relation

You can use the graph-editor in order to create relations between objects. In the next steps, you will learn how to create a relation between a company and its location and its branch.

- 1. Move the mouse cursor over **Miller plc**.
- 2. Now place the mouse cursor on the _____icon to the left, press the left mouse button, hold it down and drag the mouse cursor to **London**. Release the mouse button.

A window with the selectable relation goals will appear. Choose the required relation and confirm it with OK:





A grey arrow will appear that connects both objects:



3. Move the mouse cursor to the grey arrow: the text *organisation* is located in will appear and be visible as long as the mouse cursor is positioned on the relation arrow.



4. Switch to the Knowledge Builder. In the left menu, choose the type "Topic" and then "Sector". Then click on the tab "Objects". Drag the object **Construction Branch** into the graph-editor.

London	ρ	Subtypes Objects			≣‡⊡
FOLDER	^	Q 4 &	A	★ ④ 與 ℡	
SEMANTIC KNOWLED	GE				
Object Types		Name			^ ^
🕨 🎱 Location		Automotive Sector			
🕨 🎔 Object		Construction Branch			
🖌 🕮 Organisation		IT sector			
🏽 Company					
Institution					
Person					
Project		<			>
Topic				Sector	02
Sector		Construction Bran	ch		
Relation types		A			^
Attribute Types	- 1	Attributes			
TECHNICAL	۰.	Name	≡	Construction Branch	
 Rights 	, ~			Add attribute	
Community		Relations			
	^	soster of organisation	=	Major 9 Coor	
		sector or organisation	_		
	\sim	sector of organisation	=	Schubert KG	*

5. Connect **Miller plc** with **Construction Branch** using the relation *organisation is part of sector (Instances of sector)*:





Additionally to the relation between **London** and **Miller plc** the graph-editor now displays the newly established relation between **Miller plc** and **Construction Branch**:



3.2.2 Deleting a relation

1. Imagine you want to delete the relation between Construction Branch and Miller plc: Move the mouse cursor to the relation between **Construction Branch** and **Miller plc**.

2. Press the right mouse button on the relation between two objects: a context menu with the option **delete relation** will be displayed.

3. Click on Delete relation. A dialogue appears which asks you to choose the relation to be



deleted. Choose the relation and confirm by clicking on OK.



The relation *organisation is part of sector* between **Construction Branch** and **Miller plc** will be deleted.

3.3 End of chapter and outlook

Congratulations! You have finished the third section successfully.



Let s move on. You will get to know the following contents:

- Creating and editing new types
- Defining and using attributes
- Defining and establishing relations

We wish you lots of fun!

4 Definition of the Knowledge Graph schema

The Knowledge Graph consists of **element types (schema)** and **elements**.

The schema defines which object types, relation types and attribute types (properties) will



be elements of the Knowledge Graph - it is the structural design of the Knowledge Graph. According to this structural design which has to be composed by the user at the very beginning, the graph can be complemented successively with new **objects** with their **interrelations** and **attributes** very easily.

4.1 Creating and editing new types

The preceding sections described how to create, modify and delete **objects** (Miller plc, Mr Hans Henry) belonging to certain **types**. This section describes how to extend the graph by introducing new **types**.

	Subtypes Objects			
FOLDER	Q 4 🗞 🕲	A 🗙 🕞 🖗 🖷		
SEMANTIC KNOWLEDGE NETV				
 Object Types Location Object Object Organisation Person Project Project Relation types Attribute Types 	Name Object Organisation Person Product Project Sector Semantic Knowledge Network		^	Internal Name person
TECHNICAL	<			
Rights Trigger Registered objects	Semantic Knowle	dge Network Overview Details		
KESI	O Semantic Knowledge Netw	Properties of the type		
 View configuration Core properties 	 Location Object Organisation Person Project Yopic 	 Name Color Icon Type of Type of 	 Seman Seman Attribu Objekte 	tic Knowledge Network

Types in the Knowledge Graph are organised hierarchically:

A type is always a subtype of another type, altogether being subject to the top-level type. Each new type automatically becomes a subtype of an existing type. For this reason it is very important where you create a new type because every subtype inherits all properties (schema) of its super-type. These logical connections are a huge benefit when being considered during modelling.

For example it is not advisable to create the type **state** as a subtype of **project** because the characteristic properties of both types differ highly.

1. Create a new subtype to the type **Location**.

2. Click on the KNOWLEDGE GRAPH folder in the organizer. The type tree will be displayed.

3. Click on the grey arrow left to the type **Location**. The hierarchy view of this type opens: the subtypes (**City**, **Country**, **Federal State**) of **Location** are displayed.

$\mathcal{O}_{\mathcal{O}}$	Subtypes Objects	$\equiv :$	ŧ 🗖
FOLDER			
SEMANTIC KNOWLEDGE NETWORK			
Object Types	Continent		
	Name	^ Internal Name	^
City	City		
 Country 	Country	and the second	
Federal state	Federal state	· ·	
Object	Location		

4. Click on **Location** to see the object subtypes. Enter **Continent** in the input line of the Subtypes tab in the right half of the window and click the **D**-button. Now another *subtype* is assigned.

SEMANTIC KNOWLEDGE NETWORK
Object Types
🔺 🥥 Location
City
Continent
Country
Federal state
🕨 诃 Object
🕨 🏢 Organisation
🚨 Person
🦠 Project

4.2 Defining and using attributes

Attributes are properties or characteristic features: every object has a name. Other examples for attributes are title, password, postal code. Attributes consist of three elements:

1. the **attribute type** (e.g. date, choice, colour code), which defines the data format of the attribute values,

2. the **attribute schema**, defining what objects/types (objects of person) can be assigned with values of the attribute and

3. the **attribute value** (Mr, 64293...) as a concrete characteristic of the property belonging to type or object.

4.2.1 Defining a new attribute schema

The attribute *date of birth*, which is to be defined for the type *person*, will illustrate how to introduce new types of attributes and how to assign attribute values.

1. Click on the type person in the left part of the organizer. The *type editor* for person will be displayed in the lower half of the *Subtypes* tab.



2. Click the button Define new attribute type

A selection window will be displayed, listing the various data types in which an attribute can be stored.

Choose attribute valu	ie type
ttribute	^
oolean	
hoice	
olor value	
ate	
ate and time	
le	
exible time	
oat	
eo position	
roup	
iteger	
ternet shortcut	
terval	
assword	~

3. Choose *Date* as attribute type and confirm your choice with OK.

An editor window will be displayed for the new attribute:



👯 New attribute type			
Attribute name	date of birth		
Supertype	Attribute		
Defined for	Instances of Person		
	Instances of the type		
Internal Name			
[May have multiple occurrences		

OK Cancel

4. Enter the name **date of birth** for the new attribute in the input line *Attribute name* and confirm with OK.

NOTE:The option *May have multiple occurrences* determines whether an attribute can be assigned more than once to a single object (which makes sense, e.g. for the telephone number attribute) or not, as in the case for the attribute *date of birth*, which can only be assigned once to each person.

4.2.2 Assigning the new attribute

The attribute *date of birth* now is available to be filled with values for all objects of the type person.

4.3 Defining and establishing relations

Relations form the actual net of a Knowledge Graph. They connect types and objects with each other and provide context between objects of the Knowledge Graph. Relations consist of two components:

1. a **relation schema** (relation type), which specifies which objects/types can be connected using a relation and

2. the **actual relation** (relation object) as a real connection between two objects.

From a technical point of view, each relation schema is a type (relation type) and a specific relation between two objects in the Knowledge Graph is an object (relation object) of the relation type.

To use a relation, a relation schema is mandatory.

4.3.1 Defining a new relation schema

Using the example relation *manages project*, a relation which is defined for the object type person, you will learn how to define a new relation schema.

1. Click on the type **Person** in the left part of the organizer. The *type editor* for person will be displayed in the lower half of the tab *Subtypes*.

ρ	Subtypes Objects			≡*□
FOLDER		A X O & E		
SEMANTIC KNOWLEDGE NETWORK				5
 Object Types Location 	Name Person		Internal Name person	Ŷ
City Continent Country Federal state Golject Ganisation Project Project Ganisation Relation types	<			
TECHNICAL	Person			
Rights Trigger		Overview Details		- 0
k Registered objects * * REST	Semantic Knowledge Netw Location	relations of objects has created	Instances of User relation	^
Wiew configuration A Entire semantic network Core properties	Object Grganisation Person Project	is fallback user instance of person works for organisation	Instances of REST Service, Instances of REST Resource Instances of Organisation	
< > Community	▶ 💡 Topic	intericed relations	Define new relation type	~

2. Click the button Define new relation type . An editor window for a new relation schema will open:

Type of relation	with own inverse relation		
	Relation	Inverse relation	
Name			
Domain	Instances of Person		
Internal Name			
virtual			
		Create Cancel	

3. Fill in the name *manages project* for the new relation and *has project manager* for the inverse part of the new relation.

4. Click the button _____ next to the *Domain* input line of the *Inverse relation* (target domain).

Cancel

0	
$\bigcirc \bigcirc $	D
$\bigcirc \bigcirc $)
00)

Type of relation	with own inverse relation	~		
	Relation	Inv	verse relation	
Name	manages project	h	as project manag	ger
Domain	Instances of Person			
Internal Name				
virtual				
			Create	Cancel
An input windo	w will open. Enter project and co	nfiri	m with OK.	
			×	Defined for
En	ter the (beginning of the) type name:			Instances of Project
Project				Types of Project

In the selection window, choose **Project** as a possible target domain to permit the relation between objects of this type and objects of the type *Person*. Your cursor is back in the input window of the new relation.

5. Click the button CREATE. The schema definition of the new relation type is then complete.

4.3.2 Assigning the new relation

ОК

Cancel

As of now the relation *manages project/has project manager* is available to connect objects of *Person* and objects of *Project*.

1. Display the objects **Hans Henry** and **Construction Project School** in the Graph-Editor.

2. Draw the relation *manages project/has project manager* between **Hans Henry** and the project **Construction Project School**.

A dialogue with the possible relation types will show up. Confirm the relation by clicking on OK.



Type of relation

manages project (Instances of Project)		^
		~
	ОК	Abort

As a result, the newly built relation will be displayed as follows in the graph-editor:



4.4 End of chapter and outlook

Congratulations! You have finished the fourth section successfully.



Let s move on. You will get to know the following contents:

- Creating and displaying a search folder

- Creating a simple, a complex and a new structured query



We wish you lots of fun!

5 Searching in the Knowledge Graph

5.1 Searching objects (Tip)

So far you know how to perform a search by using the input field in the left upper corner of the organizer. In addition, the Knowledge-Builder supports complex search queries.

In the following you will learn how to create a search folder.

5.2 Creating and displaying a search folder

Structured queries are stored in search folders. You can find the *Searches* Folder in the *Work-ing Folder* in the left part of the Knowledge Builder.

In principle i-views is multilingual and adjusts to the language of your systems software.

However, the searches you create and store in the folder *Searches* keep the labellings you initially assigned. For this reason the tryout Knowledge Graph provides English as well as German labelled examples.

1. Click the arrow situated in the left part of the organizer window next to *Working Folder*. The arrowhead will point downward and the sub-folders will expand.

2. Click on the Searches folder to see its contents:

FOLDER	Organizing folder				≡×□
Working Folder (workingFolder)	A 🗟 🖻 🖉	😌 🖪 🖨 🗛	6 🖬 🖫 🐨		
Personal Data	Name	Туре	Registry key	Semantic element	Details
Private	Companies in Hesse	Structured query			
Recently accessed objects	Companies in the Construction	Structured query	A CONTRACT OF A CONTRACT OF	A CONTRACT OF A CONTRACT OF	+
Ouery results	Search Name and Synonym	Query	SearchKB	A	A CONTRACT OF
SEMANTIC KNOWLEDGE NETWORK	all employees of organisations	Structured query	•	• ·	• · · · · · · · · · · · · · · · · · · ·

5.3 Structured queries

Structured queries combine types, objects and relations. Those queries are stored in a folder to be accessed at any time. Only the search query is stored, but not the result. Hence, if the Knowledge Graph changes, the result will change accordingly.

Structured queries are ideal for formulating regular search tasks to run within the Knowledge Graph: who is responsible for which project with which partner? Who has skills in a specific area and could serve as a contact?

Some examples are already contained in the tryout Knowledge Graph. They illustrate the power of structured queries as shown in the following sections.

5.3.1 Creating a simple structured search

Create a search, analyzing the relation *person works for organisation*.



1. Go to the <i>Searches</i> folder inside the <i>Working Folder</i> and use the button Ω to create you					
own struc	tured quer	гу.			_
A dialogue	e window w	vill open:			
: Create	new			×	
Name					
all emplo	yees of orga	inisation			
Standard	Extended				
Full text qu	uery			<u>^</u>	
Query					
Search Pip	eline				
Structured	i query				
				\checkmark	

2. Name your new query **all employees of organisations** in the input line. Choose the search type *Structured query* by clicking the equivalent on the *Standard* tab. Confirm your input with OK.

OK

Cancel

3. Click the new structured query which is now listed in the folder *Searches*. The still empty query will be displayed in the right part of the organizer.



4. Click the field *Top-level type* and overwrite Top-Level type with **Person**. Confirm the entry by pressing RETURN. In the following menu you have to choose **Person** again. Confirm with OK.

5. Click the button **and choose the option** *Relations...* from the context menu. The Relations selection menu will be shown.



- 6. Choose *person works for organisation*. The simple structured query is complete.
- 7. Initiate the query by clicking the button

The search result will be displayed in the lower right part of the organizer window.



5.3.2 Creating a complex structured search

Displaying a prefabricated example

The following prefabricated example serves as an introduction to the creation of complexly structured queries: Inside the folder *Searches*, the query *companies of construction branch located in London* identifies objects of the type **Organisation**, belonging to the **Construction Branch** AND being located in **London**. The constraints of this query are being met by





combinations of types, objects, relations and attribute values.

1. To complete the circumstances for the structured query, add the relation *organisation is part of sector* with the target domain **Construction Branch** to the object **Miller plc**.

 \circ

2. Initiate the search by clicking the button

; the search results will be displayed in

the lower right section of the organizer window.

) Structured query ♀≡ Companies in the Cor	struction Sector in London	=	*
Working Folder (workingFolder) {Organizer} A Searches	+ I Organisation → Relation + P organisation is part of sector	or 💿 has Target 🖶 💡 Sector 🔍 fixed	Construction Branch	eters ^
 Companies in Hesse Companies in the Construction Sector in London Search Name and Synonym (SearchKB) 	o ^o Relation 🛨 🧬 organisation is located in	💿 has Target 🛨 💽 Location 🔍 fixed 🏾 🌖	London	
 			Query Res	,× ult≖ ☷
Recently accessed objects Query results				Ð
SEMANTIC KNOWLEDGE NETWORK	Name	^ [3] Sector	[5] Location	^
 Object Types Location 	Miller plc	Construction Branch	London	

5.3.3 Creating a new structured query

Create a new structured query to find **organisations** located in **Germany**. This structured query is to use the already existing relation *organisation is located in* and also includes the constraint of the relation s transitivity.

1. Create a new search named **companies located in Germany** inside the folder *Searches*.

2. Choose the relation *organisation is located in* and add the relation *is part of* to the target object **Location**.

NOTE:For the structured query, you just have to add the relations. The related element (here: Top-Level-Type **Location**) will be added automatically.

The query identifies **organisations**, which are connected to **Locations** via the relation *organisation is located in*. The direct locations of organisations (towns) are located in states, which for their part are located in countries. This correlation is represented by the *is part of* relation.



TIP: The *is part of* relation has to be included as often as required to consider the **Country**Germany determined in the query, beginning with **City** up to the **State**. The number of iterations of the relation *is part of* is regulated with the option *Repetitions*.

3. Activate the option *Repetitions* by clicking the button **at the left side of the relation** *is part of* in the search query:



4. The option *Repetitions* is part of the context menue, opening for the relation. Move the slidecontrol at the lower right of the input line to the right as far as possible. Now the relation will be used at least once in the query and, because of the infinity-parameters, as often as required to proceed the path **City** *is part of* **State** - **State** *is part of* **Country**... until the end.

÷.	🥜 is part of	has Targe	et 🖶 🥥 Location
	mb	Repetitions	×
		[4] is p a from	art of to
		1	~ <u>~</u>
			-
		Short	est path
		Abort	ОК

5. In order to set the location to **Germany**, click the + button beside the relation target to call up the context menue and select the options *Identify* and *Specify objects*.





	Semantic elemer	nt	
Germany			
ОК	Cancel	Parameter	

The complete structured query appears as follows:

Structured query ♀≡Companies located in Germany	≡⇔	
Image: Second Seco		^
or Relation 🐨 💣 organisation is located in 🕐 has Target 🗣 🖤 Location <u>A</u> Relation 🕂 P is part of 1	rmany	
		~
		>

The search result will be retrieved:

Structured query ₽≡Companies located in Germany	*
Image: Second Seco	^
σ ^ρ Relation 🕂 🖋 organisation is located in 💿 has Target 🕂 🕒 Location	
🔗 Relation 🚸 🧬 is part of 🛛 (1) 🧿 has Target 🖶 🥥 Location 🔍 fixed 🕥 Germa	y
	~
Image: Constraint of the second se	ult▼ 🇱
	Ð
Name ^ [3] Location	^
Administrative Court Berlin Berlin	
Glass-Henry Munich	
Meier & Sons Berlin	
Schubert KG Frankfurt	

To visualize the search results and the semantic correlations that led to the search result you can use the graph-editor.

6. Select your search results you want to visualize and open the graph-editor. The graphic display of the search results are shown as follows:



Companies located in Germany and their connections to Locations are displayed graphically. **Schubert KG** is located in **Frankfurt**, **Frankfurt** is (a geographical) part of **Hesse** and **Hesse** *is* (a geographical) *part of* **Germany**. The company **Meier & Sons** is also part of the search result, even though there is only one location, but the information for the federal state is missing. The relation *is part of* has been traversed two times to find the **Schubert KG**but only once to find **Meier & Sons**.

5.4 End of chapter and outlook



Congratulations! You have finished the fifth section successfully.

Final spurt. You will get to know the following contents:

- Building your own models - import & export pf data from external sources (with the aid of an example)

- Other Knowledge Graphs
- Contact: Question and Answer

We wish you lots of fun!

6 Building your own models

6.1 Import and Export of data from external sources

i-views provides a number of ways of importing data into a Knowledge Graph or exporting them from a graph. The representation of objects from a Knowledge Graph in a table is known as **table mapping** or simply **mapping**.

When importing, the data is read from a table and created in the Knowledge Graph as objects, types or attributes, targets of relations, etc. When exporting, objects of a Knowledge Graph and their properties (attributes, relations) are written in a table. The following data formats are supported:

- Fields in CSV format; for example from Excel files
- LDAP directories
- RDF files
- Databases with ODBC, SQL or Oracle interfaces
- XML files

6.1.1 Example: Import of persons from a CSV file

	CSV/Excel file ⊖≡ Personal Data					≣⇔⊡
Working Folder (wo			19			
Searches	Personal Data	Personal Data				
Companies ir	1: Instances of Person	CSV/Excel file Options	.og Registry			
Companies ir	2: Attribute Name Current language	Import file: C\Users	liser1\nerconal_data_csv		Sho	w table
🔎 Search Name	3: Attribute Title	Evenet file				
🔎 all employees	4: Attribute Telephone Number	export me.			Sho	w table
Companies Ic		Options				
Personal Data		Table file type CSV	file 🗸 🗸			
Private		✓ 1st row contains heading	g 🗹 Cell values			
Recently accessed o		Line March 199		Name Title	Telephone Numbe	er 👘
Query results			Separator 2	Peter Hinz Mr	040-1234-56	N
SEMANTIC KNOWLEDGE		by heading	O lab 3	Melanie Kunz Mrs	030-9876-54	
1 Object Turses		Up position	O Space 4	Rezzo Rumpel Mr	0151-5395843	
Object Types			•		_	
City		Columns: Read from data	Source			
Continent		Pesition Heading	Field length Type	Mappings	Identifier	Column ^
Country		1 Name	Variable String	2: Attribute Name Current language		Α
Federal state		2 Title	Variable String	3: Attribute Title		В
P Ubject		3 Telephone Numb	er Variable String	4: Attribute Telephone Number		С
< A See Organization >		<				>
Community		Edit Add col	umn Remove column	ns Move up Move down		
	C >					

Mappings are stored in the *Working Folder*. The example above shows the mapping for an import of a list of **Person** objects from a CSV file. The import comprises the attribute values for *Name*, *Title* and *Telephone Number*.

The file *personal_data.csv* is part of the tryout installation in order to facilitate the first steps for importing data from external sources. To perform this first import, proceed as follows:

1. Open the mapping **Personal Data** and click the button **...** next to the path to the file location.



^{CSV/Excel file}	
Personal Data	Personal Data
1: Instances of Person	CSV/Excel file Options Log Registry
 2: Attribute Name Current language 3: Attribute Title 4: Attribute Telephone Number 	Import file: C:\Users\User1\personal_data.csv Show table Export file: Show table

2. Choose the file **personal_data.csv** which is contained in the folder *resources* and confirm your choice by clicking OPEN.

File name:	personal_data.csv ~	C	SV	~	
			Open	Cancel	

3. Click the button 🕋 to start the import.

CSV/Excel file ■ = Personal Data		≡⇔⊡
Personal Data	^ Personal Data	
1: Instances of Person	CSV/Excel file Options Log Registry	

4. Check the result by clicking on the type **Person** in the left part of the organizer window in section KNOWLEDGE GRAPH.

P	Subtypes Objects		≡*□
FOLDER	د م 🛃 🔌 🕲	Q ★ (P) § 15	
🔺 🟐 Working Folder (workingFolder)			
Searches			\mathbf{a}
🝚 Personal Data	Name		<u>^</u> ^
Private	Hans Henry		
Recently accessed objects	Melanie Kunz		
Query results	Rezzo Rumpel		
SEMANTIC KNOWLEDGE NETWORK			
 Object Types 	<		>
Location			Person
🕨 🏓 Object	Melanie Kunz		
Organisation	Wieldine Ranz		
Person	Attributes		^
Person Project	Attributes	= [^
Person Project	Attributes Name	■ Melanie Kunz	
 Person Project V Topic Relation types 	Attributes Name Telephone Number 	 ■ Melanie Kunz ■ 030-9876-54 	
 Person Project Project Projec Relation types Attribute Types 	Attributes Name Telephone Number Title	■ Melanie Kunz 30-9876-54 ■ Mrs	^
 Person Project ^Q Topic <i>Calation types</i> <i>Attribute Types</i> TECHNICAL 	Attributes Name Telephone Number Title	■ Melanie Kunz 30-9876-54 ■ Mrs	^
 Person Project ² Topic <i>Relation types</i> <i>Attribute Types</i> TECHNICAL <i>Rights</i> 	Attributes Name Telephone Number Title	 Melanie Kunz 030-9876-54 Mrs Add attribute 	
 Person Project 	Attributes Name Telephone Number Title Relations	 Melanie Kunz 030-9876-54 Mrs Add attribute 	×
Person Project Project Project Relation types △ Attribute Types TECHNICAL Rights Trigger ✓ Trigger	Attributes Name Telephone Number Title Relations	 Melanie Kunz 030-9876-54 Mrs Add attribute 	×
Person Project	Attributes Name Telephone Number Title Relations	 Melanie Kunz 030-9876-54 Mrs Add attribute 	· · · · ·

6.2 Other Knowledge Graphs included in this tryout version

Beside the Knowledge Graph **i-views-tryout** there are two more examples contained in this test version:

• The Knowledge Graph music-example describes semantic objects around the topic mu-



sic. It is also used as basis for the examples in the i-views documentation.

• The Knowledge Graph **your-sandbox** is an empty graph for you to play with and build your own model from scratch according to your ideas and needs.

The initial username is "admin" in all graphs. There is no password.

6.3 Any questions or feedback?

CONGRATULATIONS!



You have finished the last section successfully.

We hope that this brief tutorial has given you an idea of the possibilities offered by a Knowledge Graph. You will find further information on our webpage.

6.4 Contact

intelligent views gmbh Julius-Reiber-Strasse 17 64293 Darmstadt Germany Tel. +49 6151 5006-0 E-Mail: support@i-views.de www.i-views.com

Your i-views-team