USAID MEDICINES, TECHNOLOGIES, AND
PHARMACEUTICAL SERVICES (MTaPS) PROGRAM

Improved Access. Improved Services. Better Health Outcomes.

Digital Regulatory System Strenghening

Pharmadex 2

Database Guide & Learning Course

International Version

March 2022

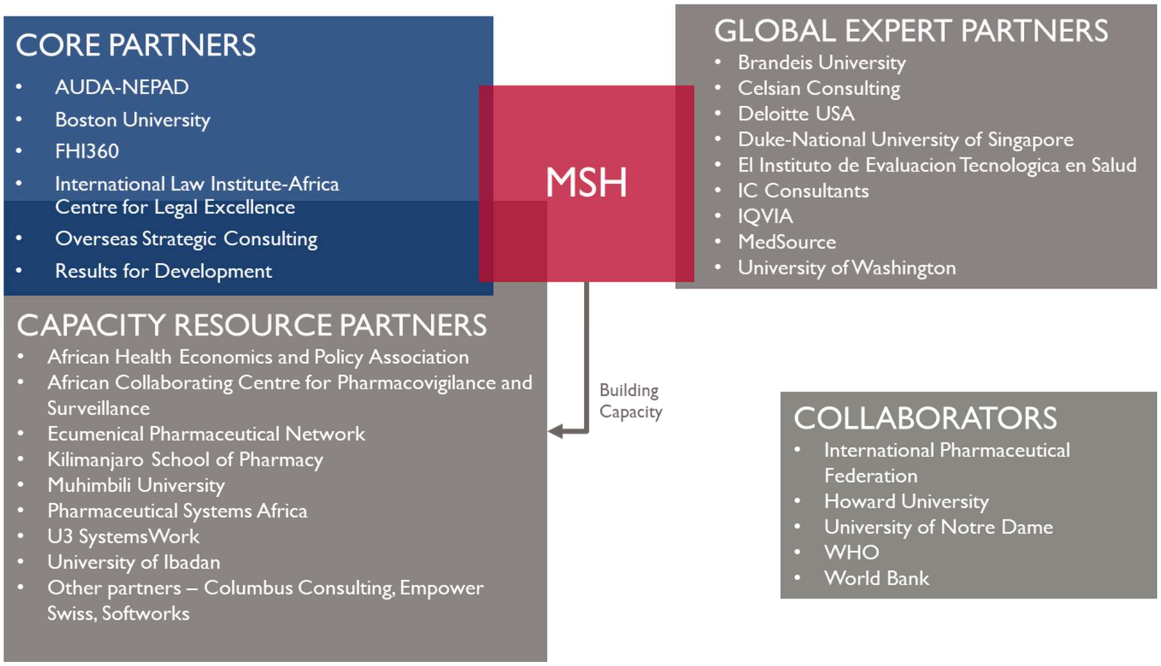
USAID
FROM THE AMERICAN PEOPLE

This document is made possible by the generous support of the American people through the US Agency for International Development (USAID) contract no. 7200AA18C00074. The contents are the responsibility of Management Sciences for Health and do not necessarily reflect the views of USAID or the United States Government.

About the USAID MTaPS Program

The USAID Medicines, Technologies, and Pharmaceutical Services (MTaPS) Program enables low- and middle-income countries to strengthen their pharmaceutical systems, which is pivotal to higher-performing health systems. MTaPS focuses on improving access to essential medical products and related services and on the appropriate use of medicines to ensure better health outcomes for all populations. The program brings expertise honed over decades of seminal pharmaceutical systems experience across more than 40 countries. The MTaPS approach builds sustainable gains in countries by including all actors in health care—government, civil society, the private sector, and academia. The program is implemented by a consortium of global and local partners and led by Management Sciences for Health (MSH), a global health nonprofit.

The MTaPS Consortium



CORE PARTNERS

- AUDA-NEPAD
- Boston University
- FHI360
- International Law Institute-Africa Centre for Legal Excellence
- Overseas Strategic Consulting
- Results for Development

MSH

GLOBAL EXPERT PARTNERS

- Brandeis University
- Celsian Consulting
- Deloitte USA
- Duke-National University of Singapore
- El Instituto de Evaluacion Tecnologica en Salud
- IC Consultants
- IQVIA
- MedSource
- University of Washington

CAPACITY RESOURCE PARTNERS

- African Health Economics and Policy Association
- African Collaborating Centre for Pharmacovigilance and Surveillance
- Ecumenical Pharmaceutical Network
- Kilimanjaro School of Pharmacy
- Muhimbili University
- Pharmaceutical Systems Africa
- U3 SystemsWork
- University of Ibadan
- Other partners – Columbus Consulting, Empower Swiss, Softworks

COLLABORATORS

- International Pharmaceutical Federation
- Howard University
- University of Notre Dame
- WHO
- World Bank

This document is submitted by:

Kim Hoppenworth
Senior Technical Adviser, PMIS
Management Sciences for Health
4301 North Fairfax Drive
Arlington, VA, 22203
Email: khoppenworth@mtapsprogram.org

Acronyms and Abbreviations

API	Application Program Interface
HL7	A set of international standards used to transfer and share data between various healthcare providers. More specifically, HL7 helps bridge the gap between health IT applications and makes sharing healthcare data easier and more efficient when compared to older methods.
NMRA	National Medicine Regulatory Authority
MTaPS	Medicines, Technologies, and Pharmaceutical Services
URL	Universal Resource Locator
USAID	US Agency for International Development
NMRA Supervisor	The supervisor is responsible for the continuous adaption of Pharmadex 2 to changing needs and the management of NMRA users. The supervisor should know well all NMRA functions. A computer science qualification is appreciated, however, not required.

Contents

PROJECT SUMMARY

Program Name:		USAID Medicines, Technologies, and Pharmaceutical Services (MTaPS) Program
Activity Start Date And End Date:		September 20, 20–September 19, 2023
Name of Prime Implementing Partner:		Management Sciences for Health
Contract Number:		7200AA18C00074
MTaPS Partners	**Core Partners**	Boston University, FHI 360, Overseas Strategic Consulting, Results for Development, International Law Institute-Africa Centre for Legal Excellence, NEPAD
	Global Expert Partners	Brandeis University, Deloitte USA, Duke-National University of Singapore, El Instituto de Evaluacion Technologica en Salud, ePath, IC Consultants, Imperial Health Sciences, MedSource, QuintilesIMS, University of Washington
	Capacity Resource Partners	African Health Economics and Policy Association, Ecumenical Pharmaceutical Network, U3 SystemsWork, University of Ibadan, University of Ghana's World Health Organizations (WHO) Pharmacovigilance Collaborating Center, Kilimanjaro School of Pharmacy, Muhimbili University, Pharmaceutical Systems Africa
	Collaborators	International Pharmaceutical Federation, Howard University, University of Notre Dame, WHO, World Bank

BACKGROUND

This course intends for programmers. Annex 1 contains SQL codes mentioned in the modules, Annex 2 answers to quizzes questions. All modules are mandatory. To execute modules, it is necessary to access the database. The preferable is local access. The latest database dump is available on GitHub https://github.com/MSH/Pharmadex2/tree/main/database. Alternatively, you ask MSH for it.

OBJECTIVE

MODULE I. INTRODUCTION

OBJECTIVE

The general introduction to the course.

SUMMARY

The Pharmadex 2 software does not need programmer's interventions to adapt to most customers' needs and requirements. It is highly configurable software grounded on the latest industry standards and best practices.

However, to implement unique customer-specific particularities or to make data available to the Data Warehouse it will be necessary to build a set of APIs or to query Pharmadex 2 database. It is a task for a qualified programmer.

To resolve this task most efficiently, a programmer should consider the following points of view to the database:

1. The data usage point of view:
   1.1. The whole data is a forest of data trees
   1.2. Each data tree root may be found by the URL string, like "pharmacy.site", "organization.authority", etc.
   1.3. Any tree is fully responsible for a data object, i.e., application configuration, dictionary, resource, organization structure, etc.
   1.4. Any data object may be found from the root of the tree to which this object belongs.
2. The API development point of view:
   2.1. Data access software services
   2.2. Data access CRUD repositories
   2.3. Object-Relation Mapped (ORM Hibernate) Java classes
   2.4. The MySQL managed relational database and SQL queries

The ability to SQL query the MySQL database is mandatory. The data access services, as well as ORM Hibernate Java classes, will be useful for Java or Kotlin programmers.

The Pharmadex 2 software codes are open. The database access codes are in a separate software library (project) "pdxmodel" Thus, Java and Kotlin programmers may use this library directly.

The main Pharmadex 2 software project “pharmadex2” is a classic Java Spring Boot REST API project. All APIs are JSON-backed. Thus, Java and Kotlin programmers may use this project as an example. In addition, it is possible to use these APIs directly from others software.

The Pharmadex 2 database “pdx2” contains Pharmadex 1 tables, Pharmadex 2 tables, views, and stored procedures. The Pharmadex 2 tables are listed as ORM Hibernate entities in package “org.msh.pdex2.model”. This package can be found in the project “pdxmodel”. All views and stored procedures in the database “pdx2” are solely for Pharmadex 2. Thus, a programmer can use them, regardless of the programming language.

QUIZ

1) The Pharmadex 2 software codes are open. What is the recommended way to adapt it to the needs of the particular NMRA?
   a) Re-programming some source codes
   b) Configuring Pharmadex 2
   c) Configuring Pharmadex 2 and building particular APIs if it will be necessary
2) A dictionary in Pharmadex 2 is a data element represented as a graph of any reasonable depth. Which point of view is it?
   a) Data usage
   b) API development
3) I'm a .NET programmer. How can I use project "pdxmodel"?
   a) As a software library
   b) For reference only
4) Is it possible to use Pharmadex 2 software API directly from the PHP application?
   a) Yes
   b) No
5) We plan to implement the Microsoft Azure data warehouse. What is the best way to access Pharmadex 2 data from ETL (Extract, Transform, and Load) process?
   a) The usage of API calls to Pharmadex 2
   b) The usage of SQL queries, views, and stored procedures
   c) The development of special APIs

MODULE 2. THE DATA FOREST

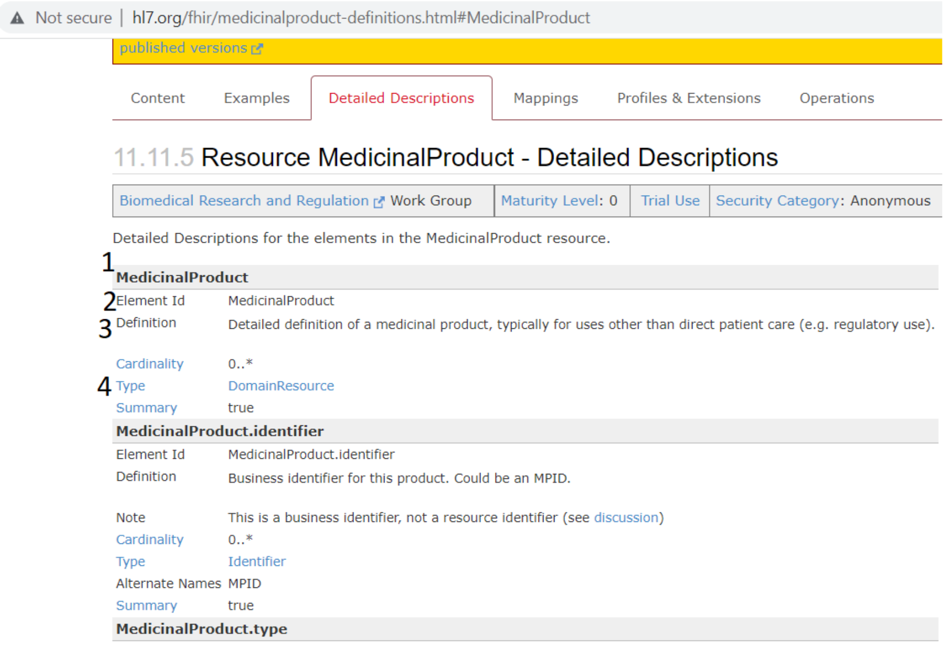
OBJECTIVE

The objective is to learn why the tree logical model has been selected for Pharmadex 2. The general data usage recommendations are provided.

SUMMARY

The logical representation of data as a forest of data trees is widely used. Any tree keeps self-sufficient data.

An example is FHIR HL7 standard implementation. The quote from them (Figure 1) demonstrates the root of the Medicinal Product[1] data tree and two branches of it – Identifier and Type.



11.11.5 Resource MedicinalProduct - Detailed Descriptions

Biomedical Research and Regulation Work Group	Maturity Level: 0	Trial Use	Security Category: Anonymous

Detailed Descriptions for the elements in the MedicinalProduct resource.

MedicinalProduct	
Element Id	MedicinalProduct
Definition	Detailed definition of a medicinal product, typically for uses other than direct patient care (e.g. regulatory use).
Cardinality	0..*
Type	DomainResource
Summary	true
MedicinalProduct.identifier	
Element Id	MedicinalProduct.identifier
Definition	Business identifier for this product. Could be an MPID.
Note	This is a business identifier, not a resource identifier (see discussion)
Cardinality	0..*
Type	Identifier
Alternate Names	MPID
Summary	true
MedicinalProduct.type	

Figure 1. FHIR quote

The data organization in FHIR HL7 is compatible with SKOS standard by W3 consortium[2]:

The Pharmadex 2 logical data structure is compatible with SKOS standard i.e.,

- Each data element is represented by SKOS originated concept[3]
- Any concept is included in a tree of concepts
- A concept contains an identifier (the element ID), a preferred label, and a description

[1] This particular example is still undergoing development and review by the appropriate Workgroups. At this time, is considered only as a draft resource not suitable for production implementation. However, for Pharmadex 2 it is the most appropriative element from HL7

[2] https://www.w3.org/TR/skos-reference/#notations

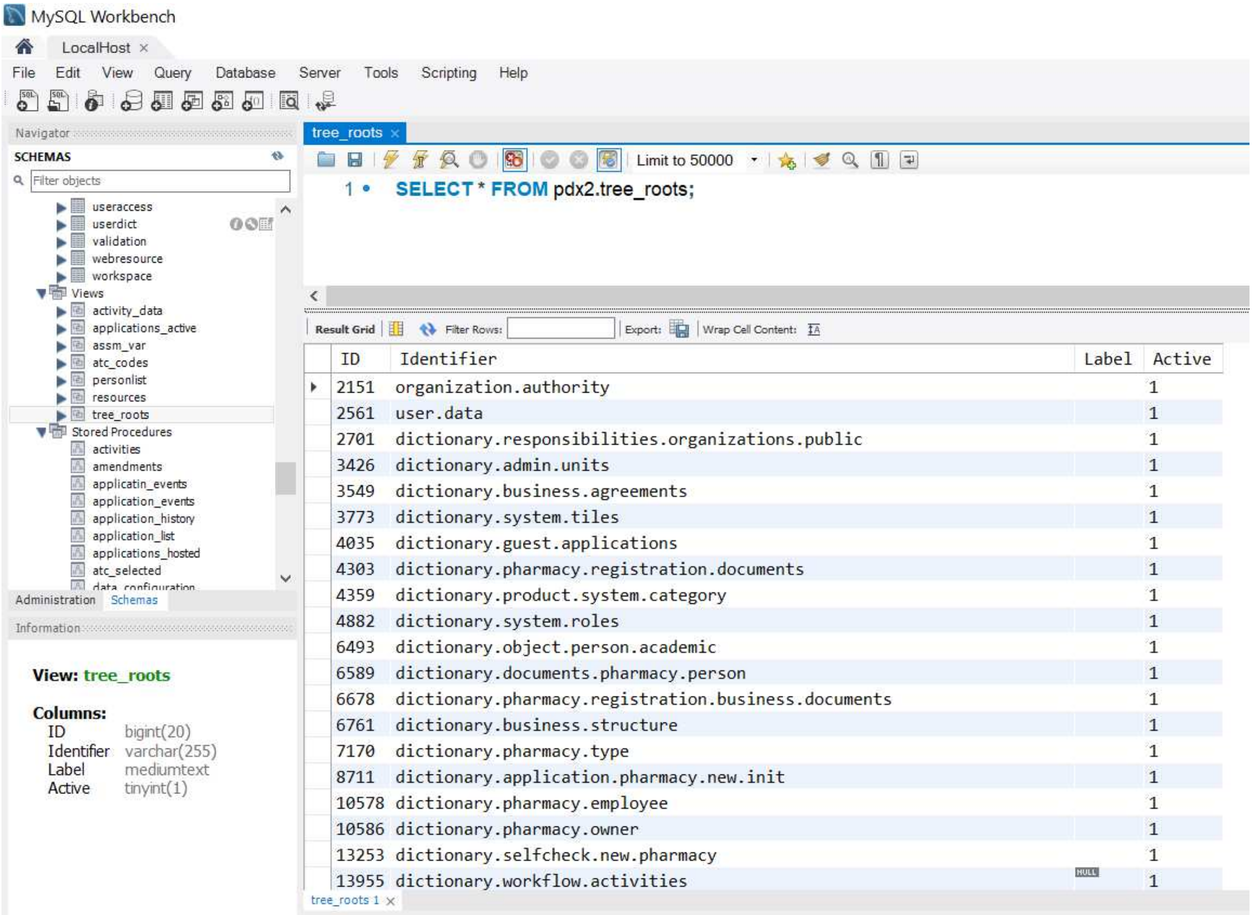
[3] https://www.w3.org/TR/skos-reference/#concepts

As in FHIR HL7, the root of any data tree is identified by a URI (Universal Resource Identifier). The URI implementation is a string in a dot-separated URL format. Examples are:

- organization.authority
- dictionary.responsibilities.organizations.public
- user.data

The usage of FHIR rules http://hl7.org/fhir/fhirpath.html to define URLs that do not list in the FHIR is highly appreciated.

The full list of the roots is available using the view “tree_roots” (Figure 2). The assignment of most URLs is a responsibility of a Pharmadex 2 Supervisor user.



```
1 • SELECT * FROM pdx2.tree_roots;
```

ID	Identifier	Label	Active
2151	organization.authority		1
2561	user.data		1
2701	dictionary.responsibilities.organizations.public		1
3426	dictionary.admin.units		1
3549	dictionary.business.agreements		1
3773	dictionary.system.tiles		1
4035	dictionary.guest.applications		1
4303	dictionary.pharmacy.registration.documents		1
4359	dictionary.product.system.category		1
4882	dictionary.system.roles		1
6493	dictionary.object.person.academic		1
6589	dictionary.documents.pharmacy.person		1
6678	dictionary.pharmacy.registration.business.documents		1
6761	dictionary.business.structure		1
7170	dictionary.pharmacy.type		1
8711	dictionary.application.pharmacy.new.init		1
10578	dictionary.pharmacy.employee		1
10586	dictionary.pharmacy.owner		1
13253	dictionary.selfcheck.new.pharmacy		1
13955	dictionary.workflow.activities	NULL	1

View: tree_roots

Columns:

ID	bigint(20)
Identifier	varchar(255)
Label	mediumtext
Active	tinyint(1)

Figure 2. The forest

The administrate features use URLs widely

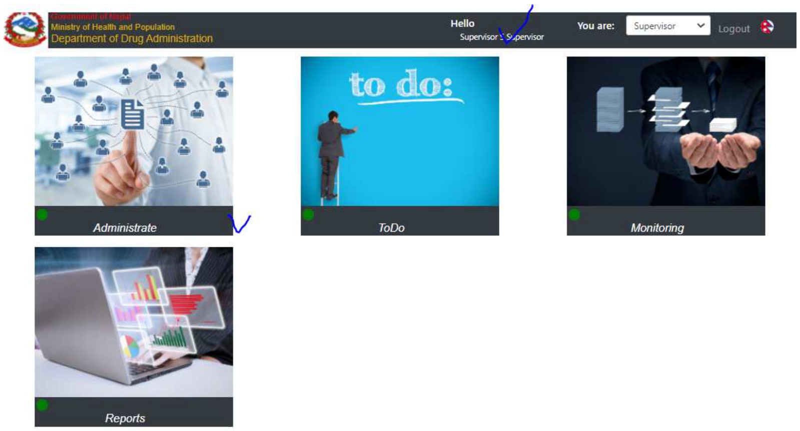


Figure 3. Administrate feature access

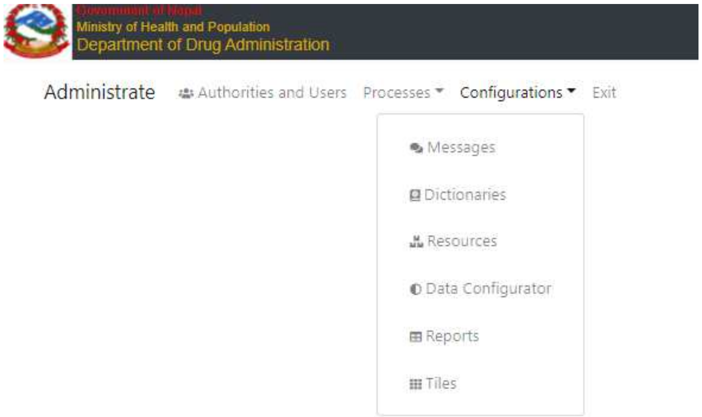
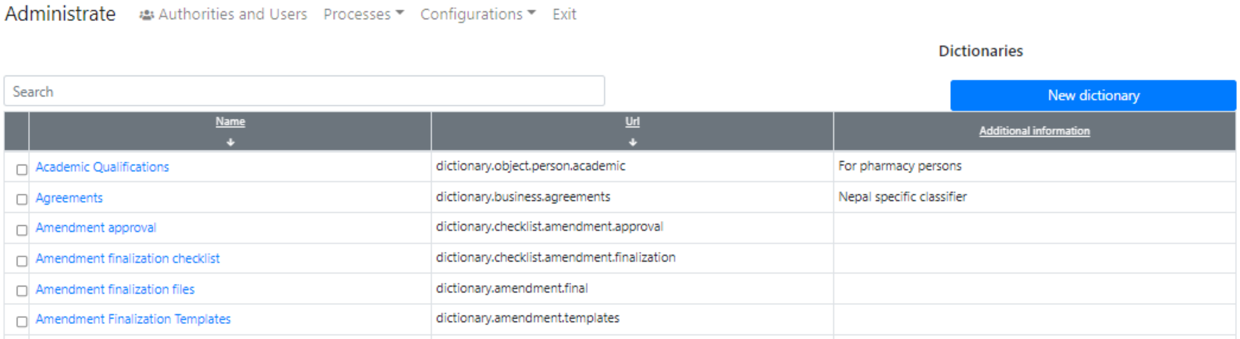


Figure 4. The configuration's feature

An example is a "Dictionaries" feature.



Dictionaries

	Name	Url	Additional information
□	Academic Qualifications	dictionary.object.person.academic	For pharmacy persons
□	Agreements	dictionary.business.agreements	Nepal specific classifier
□	Amendment approval	dictionary.checklist.amendment.approval	
□	Amendment finalization checklist	dictionary.checklist.amendment.finalization	
□	Amendment finalization files	dictionary.amendment.final	
□	Amendment Finalization Templates	dictionary.amendment.templates	

Figure 5. Dictionaries

QUIZ

1) The HL7 MedicinalProduct tree should be considered as a draft. Is it possible to use it in Pharmadex 2?
   a) No, it is impossible, because Pharmadex 2 should strictly obey the current HL7
   b) Yes, it is possible, because it is recommended to use the HL7 approach in case the HL7 standard is not available yet.
2) Administrative feature “Resources” shows URLs. However, it is impossible to find these URLs in the view “tree_root”. What do you think why?
   a) These URLs are branches in some other trees
   b) These URLs do not belong to any tree

Module 3. Data tree implementation and exploring

Objective

The objective is to study data tree implementation in the ORM and the database. The data exploration examples are provided.

Summary

The Pharmadex 2 database is implemented on the MySQL relation database engine. The name of the schema is “pdx2”. The URLs tree has been implemented using the “closure table” pattern (https://coderwall.com/p/lixing/closure-tables-for-browsing-trees-in-sql).

The “Closure Table” pattern is built on Concept and Closure ORM objects (Figure 6).

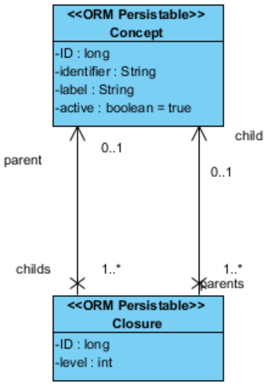


Figure 6. ORM implementation of the "closure table" database pattern

These ORM objects are backed by the database tables “concept” and “closure”.

Concept

Field	Description	Data samples
ID	Primary key	2155
Identifier	Unique Identifier of this concept inside the branch. It may be URL, language tag, variable name, etc	en_us
Label	Any textual information related to this concept	Ministry of Health and Population
Active	Boolean value. Is this concept active?	true

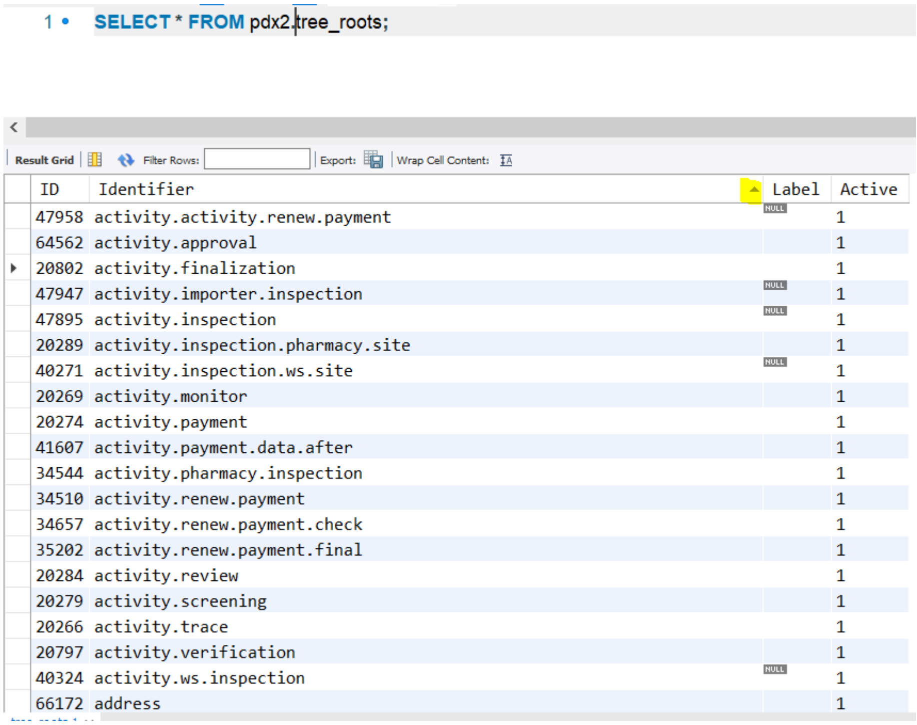
Closure

Field	Description	Data samples
ID	Primary key	7510
ChildID	Point to child concept	2155
ParentID	Point to parent concept	2154
Level	Tree level	1

To support the trees on the service level, Pharmadex 2 provides the software implemented by org.msh.pharmadex2.service.r2.ClosureService.java. It can be found in the “pharmadex2” project.

The Pharmadex 2 provides views and stored procedures to work with the trees directly, by SQL. It is possible to explore a tree from the root or any branch or leaf.

View “tree_root” selects roots A root URL is a concept for which the parent concept doesn`t exist. (Figure 7).



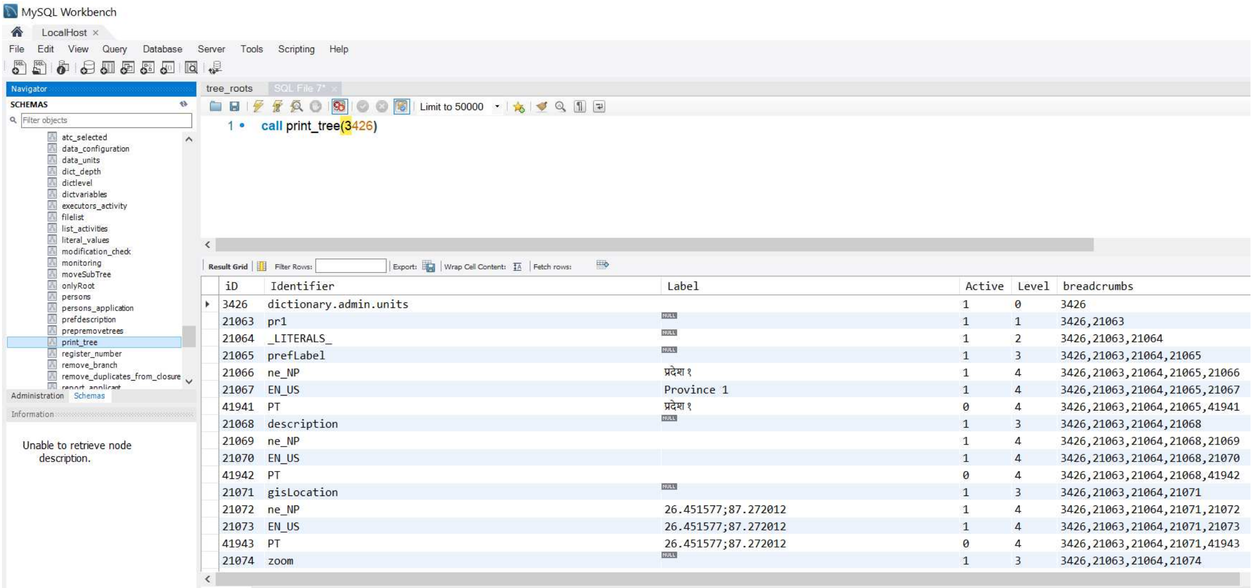
```
SELECT * FROM pdx2.tree_roots;
```

ID	Identifier	Label	Active
47958	activity.activity.renew.payment	NULL	1
64562	activity.approval		1
20802	activity.finalization		1
47947	activity.importer.inspection	NULL	1
47895	activity.inspection	NULL	1
20289	activity.inspection.pharmacy.site		1
40271	activity.inspection.ws.site	NULL	1
20269	activity.monitor		1
20274	activity.payment		1
41607	activity.payment.data.after		1
34544	activity.pharmacy.inspection		1
34510	activity.renew.payment		1
34657	activity.renew.payment.check		1
35202	activity.renew.payment.final		1
20284	activity.review		1
20279	activity.screening		1
20266	activity.trace		1
20797	activity.verification		1
40324	activity.ws.inspection	NULL	1
66172	address		1

Figure 7. Tree roots sorted by URL

The stored procedure “print_tree” allows exploring any “tree” in the “forest”. The parameter of this procedure is a primary key (ID) of the concept.

Example 1. The tree exploration from the root. The tree root is “dictionary.admin.units”.



Unable to retrieve node description.

```
1 • call print_tree(3426)
```

iD	Identifier	Label	Active	Level	breadcrumbs
3426	dictionary.admin.units		1	0	3426
21063	pr1	NULL	1	1	3426,21063
21064	_LITERALS_	NULL	1	2	3426,21063,21064
21065	prefLabel	NULL	1	3	3426,21063,21064,21065
21066	ne_NP	प्रदेश १	1	4	3426,21063,21064,21065,21066
21067	EN_US	Province 1	1	4	3426,21063,21064,21065,21067
41941	PT	प्रदेश १	0	4	3426,21063,21064,21065,41941
21068	description	NULL	1	3	3426,21063,21064,21068
21069	ne_NP		1	4	3426,21063,21064,21068,21069
21070	EN_US		1	4	3426,21063,21064,21068,21070
41942	PT		0	4	3426,21063,21064,21068,41942
21071	gisLocation	NULL	1	3	3426,21063,21064,21071
21072	ne_NP	26.451577;87.272012	1	4	3426,21063,21064,21071,21072
21073	EN_US	26.451577;87.272012	1	4	3426,21063,21064,21071,21073
41943	PT	26.451577;87.272012	0	4	3426,21063,21064,21071,41943
21074	zoom	NULL	1	3	3426,21063,21064,21074

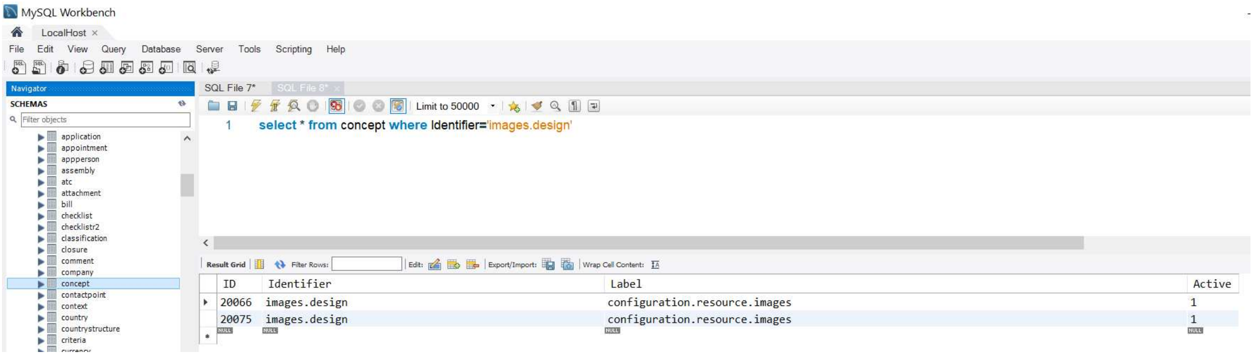
Figure 8. Explore the dictionary tree from the root

The columns are from the “concept” table. The latest column “breadcrumbs” displays the full path to the root. These are IDs of the concepts.

1. The concept 3426 is the tree root of the dictionary under URL “dictionary.admin.units”
2. Under it is 21063 is a concept “pr1”. It is a branch of the tree.
3. Under it is a concept 21064 “_LITERALS_”. It is a branch of multi-language labels
4. Under it is a concept 21065 “prefLabel”. It is a label name
5. Under it are concepts contain values of the “prefLabel” on different languages

Example 2. The tree exploration from a branch or a leaf. The branch is the “images.design” resource[4].

It is not the root of a tree, thus first we need to determine the ID of the related concept (Figure 9). There are two concepts with the same URL because the resource is created for two languages. Thus, these concepts are in separate branches.

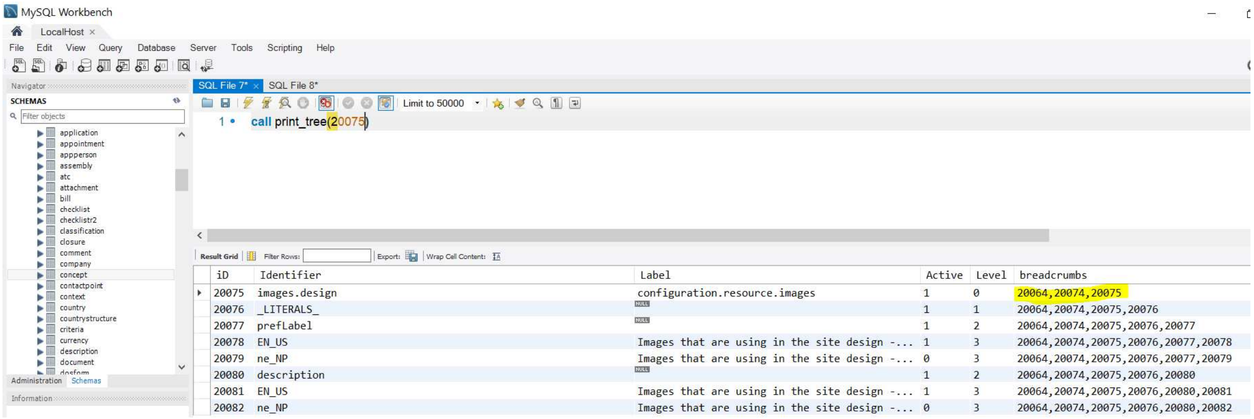


```
select * from concept where Identifier='images.design'
```

ID	Identifier	Label	Active
20066	images.design	configuration.resource.images	1
20075	images.design	configuration.resource.images	1

Figure 9. Find the ID of the branch inside the tree

Then, will get a path to the tree root for any of the concepts above (Figure 9), using the “print_tree” stored procedure. The second concept (2074) in this path should be the language of the branch.



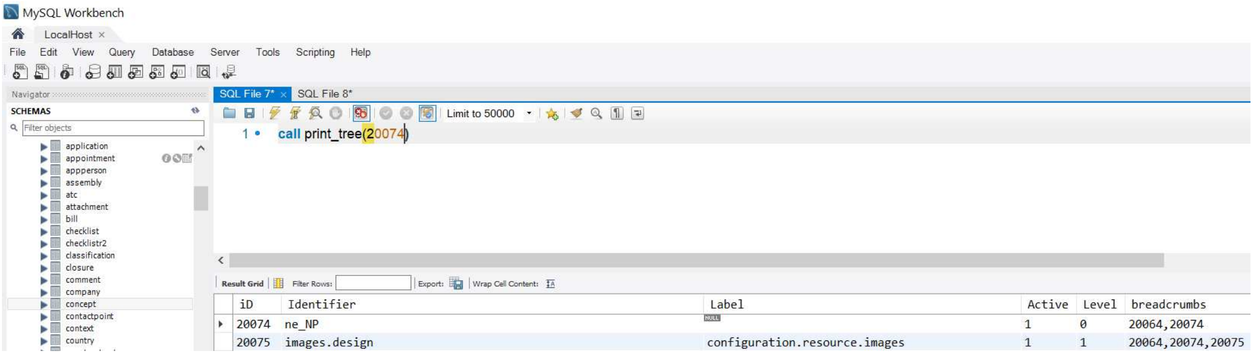
```
call print_tree(20075)
```

iD	Identifier	Label	Active	Level	breadcrumbs
20075	images.design	configuration.resource.images	1	0	20064,20074,20075
20076	_LITERALS_	NULL	1	1	20064,20074,20075,20076
20077	prefLabel	NULL	1	2	20064,20074,20075,20076,20077
20078	EN_US	Images that are using in the site design -...	1	3	20064,20074,20075,20076,20077,20078
20079	ne_NP	Images that are using in the site design -...	0	3	20064,20074,20075,20076,20077,20079
20080	description	NULL	1	2	20064,20074,20075,20076,20080
20081	EN_US	Images that are using in the site design -...	1	3	20064,20074,20075,20076,20080,20081
20082	ne_NP	Images that are using in the site design -...	0	3	20064,20074,20075,20076,20080,20082

Figure 10. Find a branch inside the tree

[4] This resource contains images that are used by Pharmadex 2 screens.

Then, will explore the concept 2074, using print_tree (Figure 11). It is the Nepali language resource.



```
call print_tree(20074)
```

iD	Identifier	Label	Active	Level	breadcrumbs
20074	ne_NP	NULL	1	0	20064,20074
20075	images.design	configuration.resource.images	1	1	20064,20074,20075

Figure 11. Find a branch of the immediate parent

Additional useful stored procedures to directly manage a tree and the forest are:

- remove_branch() – it is a correct way to remove a node in a tree. Deletion of a concept directly is not a good choice.
- moveSubTree() provides a possibility to move a branch from one tree to another.

The rest of the stored procedures are used by Pharmadex 2 software. The result of execution of them is temporary tables that may be used in SQL queries and inside other stored procedures.

It is forbidden to make changes in the procedures provided. Create and use copies instead.

Quiz

1) Why the field “Level” in the “closure” table is useful?
   a) This field is excess
   b) This field allows avoiding recursive SQL
   c) This field allows getting a root of a tree from any branch or leaf in a simple SQL query
2) Is it possible to use only the “tree_root” view to explore a tree?
   a) Yes
   b) No
3) Is it possible to explore a tree from a leaf?
   a) Yes
   b) No

Module 4. The tree implementation of multi-language Strings, Literals, Numbers, Booleans, Dates

Objective

The Pharmadex 2 provides multi-language data storage. In most cases, the data is stored for the language selected by a user in the User Interface. This feature is implemented using the data tree.

Summary

The Pharmadex 2 provides a possibility to include to the application data the common use data types such are Strings, Literals[5], Numbers, Booleans, and Dates. This data will be stored separately for the language selected by a user.

The database structure to manage them is a tree[6]. Below is an example of the tree from the "pharmacy.site" root.

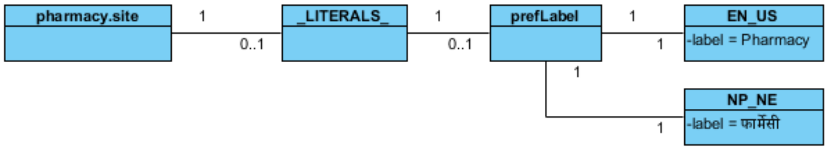


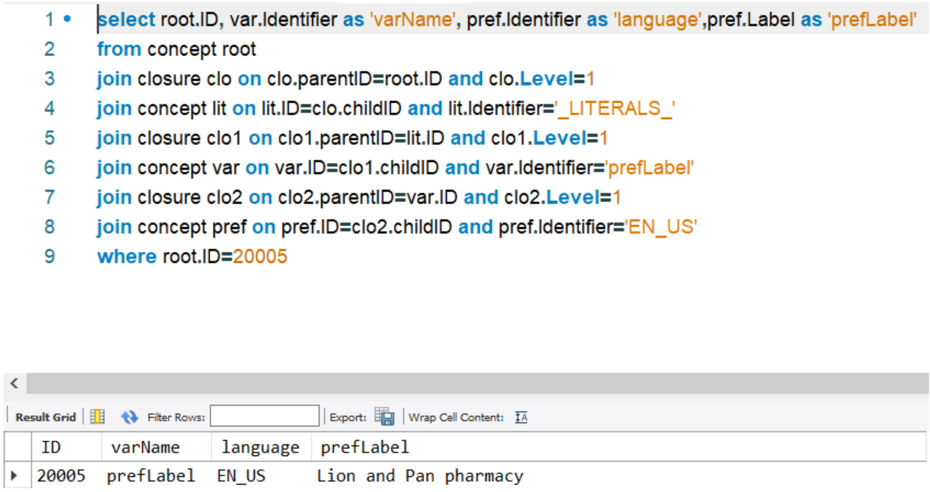
Figure 12. A literal with name prefLabel is multi-language

- The root concept with Identifier "pharmacy.site" is linked to zero or one concept with identifier _LITERALS_
- A concept _LITERAL_ is linked to a concept with identifier "prefLabel"
- A concept with the identifier prefLabel (name of the data variable) is linked to concepts with language tags identifiers
- Each language tag keeps a value in the language defined by the field "identifier".

[5] Strings and Literals are representations of a textual strings. However, Strings unlike literals do not support the multi-language feature

[6] The ORM diagram is unclear for this case - Figure 6

To explain the considerations above, it is possible to create a SQL[7] to extract values of the "prefLabel" field linked to the root concept with ID=20005. The language is en_US.



```
select root.ID, var.Identifier as 'varName', pref.Identifier as 'language',pref.Label as 'prefLabel'
from concept root
join closure clo on clo.parentID=root.ID and clo.Level=1
join concept lit on lit.ID=clo.childID and lit.Identifier='_LITERALS_'
join closure clo1 on clo1.parentID=lit.ID and clo1.Level=1
join concept var on var.ID=clo1.childID and var.Identifier='prefLabel'
join closure clo2 on clo2.parentID=var.ID and clo2.Level=1
join concept pref on pref.ID=clo2.childID and pref.Identifier='EN_US'
where root.ID=20005
```

```
ID      varName    language   prefLabel
20005   prefLabel  EN_US      Lion and Pan pharmacy
```

Figure 13. Select a value of prefLabel in en_US

[7] This query is for learning purpose only. The real query is simplest. See Annex 1.

QUIZ

1) Suppose that in some country is in use Gregorian and traditional calendar. Is it possible to keep dates separately?
   a) Yes
   b) No
2) Why does Pharmadex 2 use a concept with identifier _LITERALS_?
   a) It is the useless concept
   b) To distinct common use variables from others
   c) To improve the performance of SQL queries

MODULE 5. THE CONFIGURATION OF THE APPLICATION DATA.

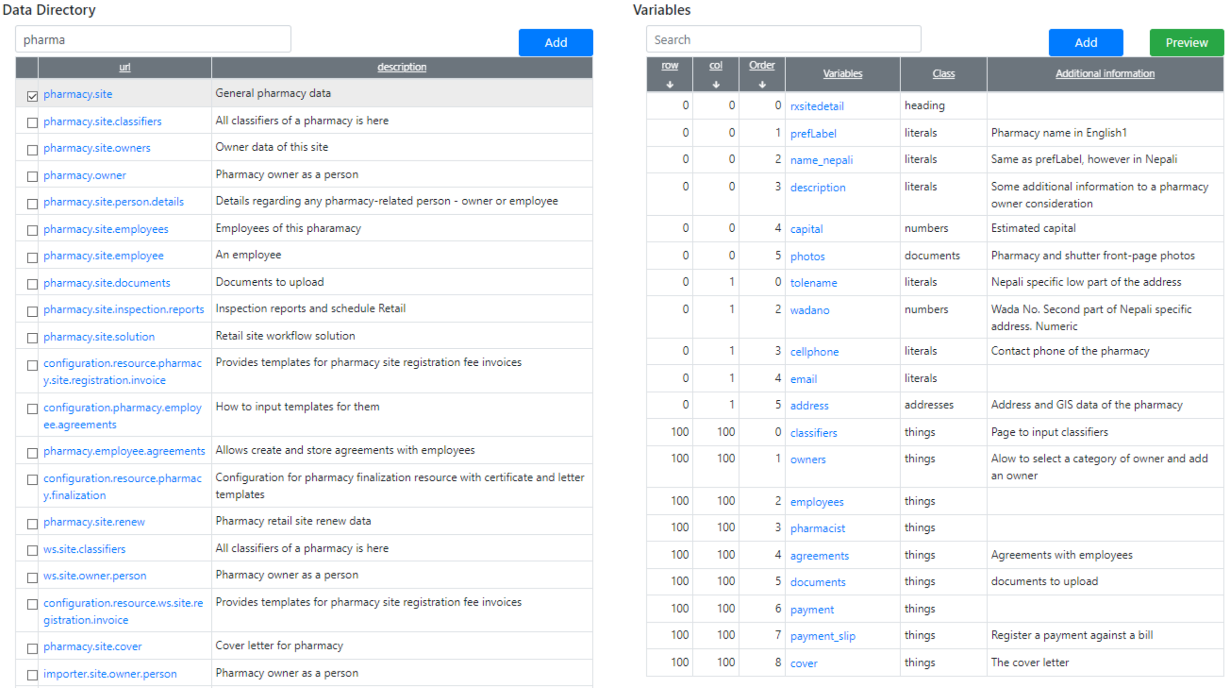
OBJECTIVE

The Pharmadex 2 uses the data tree to keep application configuration data. This data is necessary for APIs creation.

SUMMARY

The Pharmadex 2 allows the Supervisor to configure application data as well as on-screen forms.

Below, is the "pharmacy.site" application data configuration.



	url	description
☑	pharmacy.site	General pharmacy data
☐	pharmacy.site.classifiers	All classifiers of a pharmacy is here
☐	pharmacy.site.owners	Owner data of this site
☐	pharmacy.owner	Pharmacy owner as a person
☐	pharmacy.site.person.details	Details regarding any pharmacy-related person - owner or employee
☐	pharmacy.site.employees	Employees of this pharamacy
☐	pharmacy.site.employee	An employee
☐	pharmacy.site.documents	Documents to upload
☐	pharmacy.site.inspection.reports	Inspection reports and schedule Retail
☐	pharmacy.site.solution	Retail site workflow solution
☐	configuration.resource.pharmacy.site.registration.invoice	Provides templates for pharmacy site registration fee invoices
☐	configuration.pharmacy.employee.agreements	How to input templates for them
☐	pharmacy.employee.agreements	Allows create and store agreements with employees
☐	configuration.resource.pharmacy.finalization	Configuration for pharmacy finalization resource with certificate and letter templates
☐	pharmacy.site.renew	Pharmacy retail site renew data
☐	ws.site.classifiers	All classifiers of a pharmacy is here
☐	ws.site.owner.person	Pharmacy owner as a person
☐	configuration.resource.ws.site.registration.invoice	Provides templates for pharmacy site registration fee invoices
☐	pharmacy.site.cover	Cover letter for pharmacy
☐	importer.site.owner.person	Pharmacy owner as a person

row ↓	col ↓	Order ↓	Variables	Class	Additional information
0	0	0	rxsitedetail	heading	
0	0	1	prefLabel	literals	Pharmacy name in English1
0	0	2	name_nepali	literals	Same as prefLabel, however in Nepali
0	0	3	description	literals	Some additional information to a pharmacy owner consideration
0	0	4	capital	numbers	Estimated capital
0	0	5	photos	documents	Pharmacy and shutter front-page photos
0	1	0	tolename	literals	Nepali specific low part of the address
0	1	2	wadano	numbers	Wada No. Second part of Nepali specific address. Numeric
0	1	3	cellphone	literals	Contact phone of the pharmacy
0	1	4	email	literals	
0	1	5	address	addresses	Address and GIS data of the pharmacy
100	100	0	classifiers	things	Page to input classifiers
100	100	1	owners	things	Alow to select a category of owner and add an owner
100	100	2	employees	things	
100	100	3	pharmacist	things	
100	100	4	agreements	things	Agreements with employees
100	100	5	documents	things	documents to upload
100	100	6	payment	things	
100	100	7	payment_slip	things	Register a payment against a bill
100	100	8	cover	things	The cover letter

Figure 14. Data Configuration example from Pharmadex 2

Each line in this configuration is stored in the database using Concept and Assembly ORM objects. The Assembly is to store the configuration data. The Concept is to collect the configuration data to the appropriate tree and to keep language specific data in the branch "_LITERALS_".

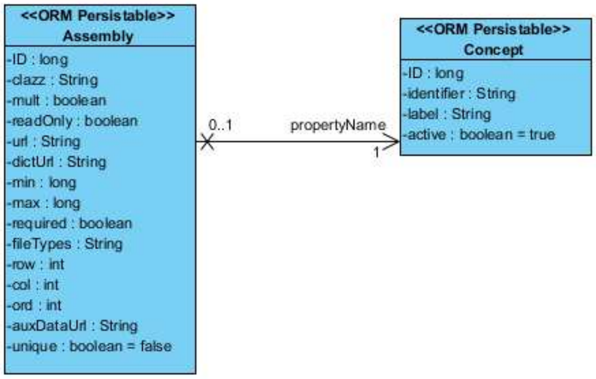


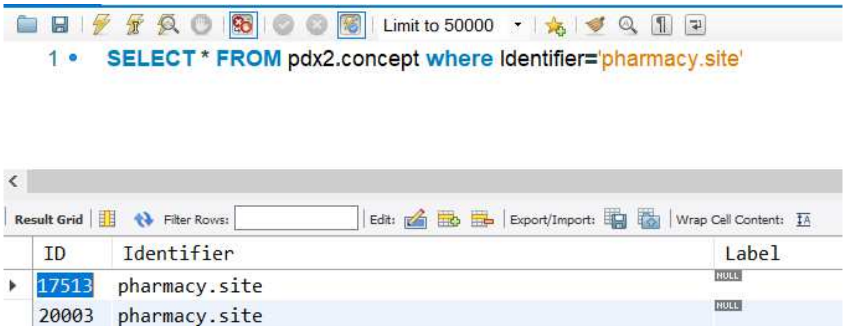
Figure 15. A line in the data configuration

The Assembly object contains many fields, because of uniformity needs. The most used of them are:

- url – data configuration URL
- clazz – the type of variable
- col, row, order – place on the screen
- required – is this data mandatory?

On the database level, the configuration data is stored to the tables “concept” and “assembly”. Thus, it is possible to create SQL queries for the configuration data.

First, will find a concept that is the root of the “pharmacy.site” configuration (Figure 16). The result is two concepts – the first is the configuration data, the second is the application data.

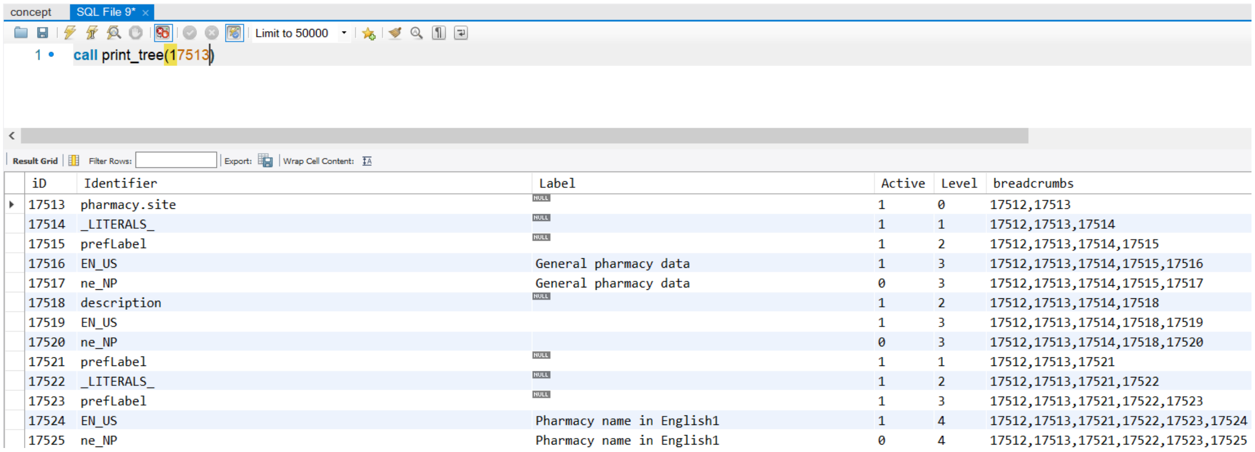


```
SELECT * FROM pdx2.concept where Identifier='pharmacy.site'
```

ID	Identifier	Label
17513	pharmacy.site	NULL
20003	pharmacy.site	NULL

Figure 16. Find a concept by the Identifier

Then, will ensure that it is the configuration data (Figure 17).

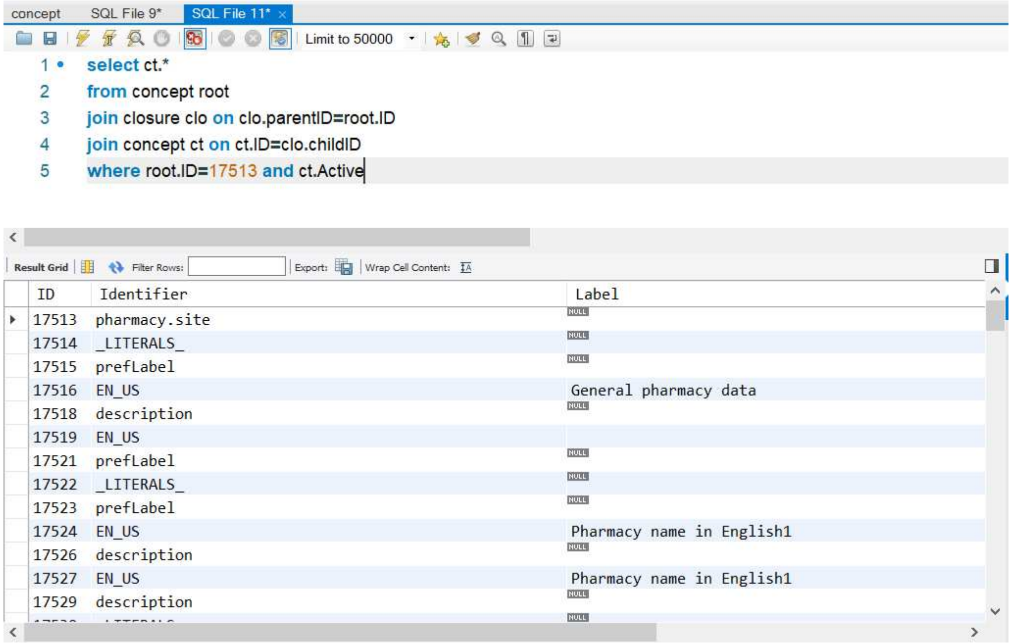


```sql
call print_tree(17513)
```

iD	Identifier	Label	Active	Level	breadcrumbs
17513	pharmacy.site	NULL	1	0	17512,17513
17514	_LITERALS_	NULL	1	1	17512,17513,17514
17515	prefLabel	NULL	1	2	17512,17513,17514,17515
17516	EN_US	General pharmacy data	1	3	17512,17513,17514,17515,17516
17517	ne_NP	General pharmacy data	0	3	17512,17513,17514,17515,17517
17518	description	NULL	1	2	17512,17513,17514,17518
17519	EN_US		1	3	17512,17513,17514,17518,17519
17520	ne_NP		0	3	17512,17513,17514,17518,17520
17521	prefLabel	NULL	1	1	17512,17513,17521
17522	_LITERALS_	NULL	1	2	17512,17513,17521,17522
17523	prefLabel	NULL	1	3	17512,17513,17521,17522,17523
17524	EN_US	Pharmacy name in English1	1	4	17512,17513,17521,17522,17523,17524
17525	ne_NP	Pharmacy name in English1	0	4	17512,17513,17521,17522,17523,17525

Figure 17. Data configuration tree

It is possible to create an SQL query that returns the same result (Figure 18)

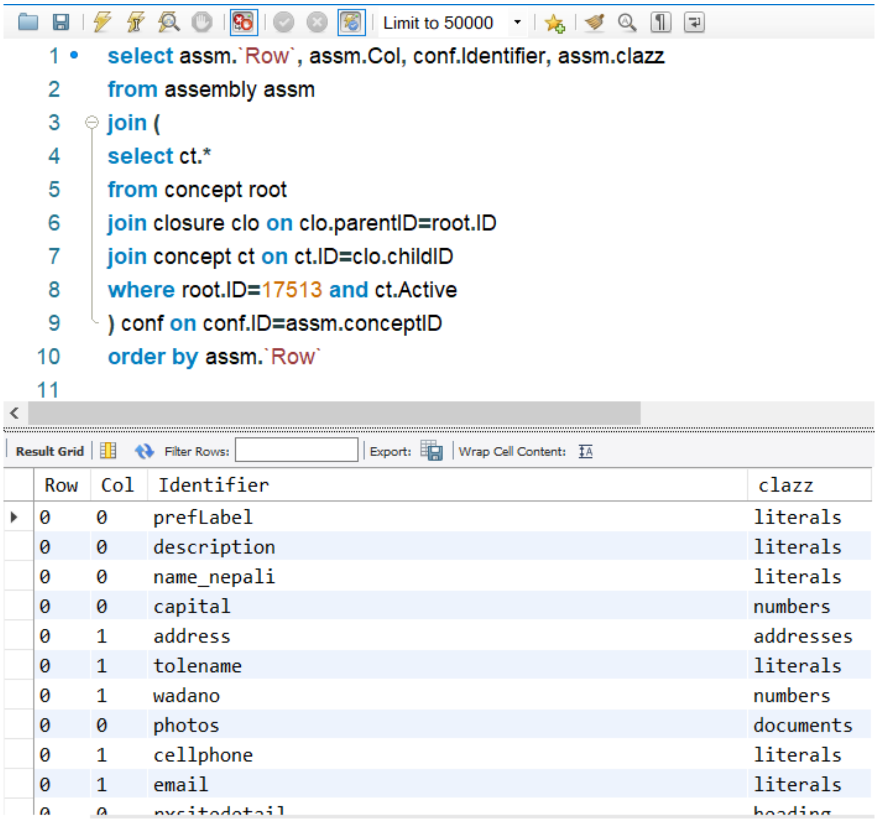


```sql
select ct.*
from concept root
join closure clo on clo.parentID=root.ID
join concept ct on ct.ID=clo.childID
where root.ID=17513 and ct.Active
```

ID	Identifier	Label
17513	pharmacy.site	NULL
17514	_LITERALS_	NULL
17515	prefLabel	NULL
17516	EN_US	General pharmacy data
17518	description	NULL
17519	EN_US	
17521	prefLabel	NULL
17522	_LITERALS_	NULL
17523	prefLabel	NULL
17524	EN_US	Pharmacy name in English1
17526	description	NULL
17527	EN_US	Pharmacy name in English1
17529	description	NULL

Figure 18. Data configuration selection use plain SQL select

Then will create a SQL to select the data configurations from the “assembly” table (Figure 19). Internal SQL is from the example above (Figure 18). The “clazz” column means the data type.



```
select assm.`Row`, assm.Col, conf.Identifier, assm.clazz
from assembly assm
join (
select ct.*
from concept root
join closure clo on clo.parentID=root.ID
join concept ct on ct.ID=clo.childID
where root.ID=17513 and ct.Active
) conf on conf.ID=assm.conceptID
order by assm.`Row`
```

Row	Col	Identifier	clazz
0	0	prefLabel	literals
0	0	description	literals
0	0	name_nepali	literals
0	0	capital	numbers
0	1	address	addresses
0	1	tolename	literals
0	1	wadano	numbers
0	0	photos	documents
0	1	cellphone	literals
0	1	email	literals
0	0	pxsitedetail	heading

Figure 19. Select data configuration using SQL

QUIZ

1) Who is responsible for application data configuration?
    a) Supervisor
    b) Programmer
    c) Both
2) Which table are not in use for data configuration?
    a) concept
    b) closure
    c) activity
    d) assembly
3) Does the “clazz” column is available for the Supervisor?
    a) Yes
    b) No

MODULE 6. THE APPLICATION DATA

OBJECTIVE

This module explains how the application data may be found, and, then, accessed by SQL.

SUMMARY

To collaborate with NMRA the Business User should create the electronic application data and, then, send it to NMRA. The data structure of the application data has been defined by the supervisor. The application data is stored in the tree. The root concept of this tree is the application's URL. Below is an example for "pharmacy.site" application data.

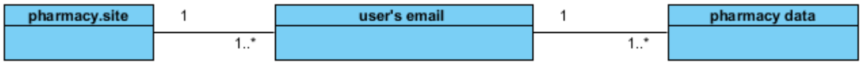


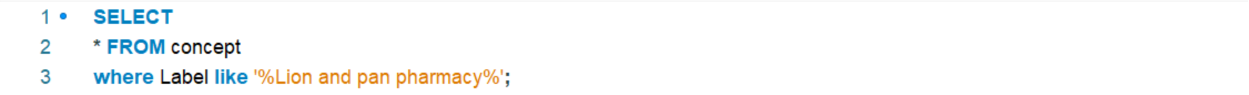
Figure 20. How to application data is stored in a tree?

1. The concept "pharmacy.site" is the root of data of the "pharmacy. site" applications
2. It is linked to many concepts of applicants' emails – owners of applications
3. Email of each applicant is linked to one or many applications data concepts (typically one)

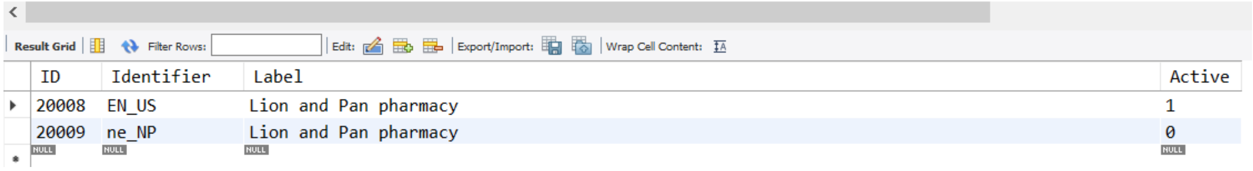
Thus, the data of any given application may be found using the root concept of the application. The structure of the application data is defined in the configuration. Because the configuration is uniform, the data of any application can be gotten by the same SQL.

There are many approaches how to finding the root concept. Suppose, we will need "Lion and Pan pharmacy".

First, select concepts with this label "Lion and Pan pharmacy". The concepts for two languages will be selected.



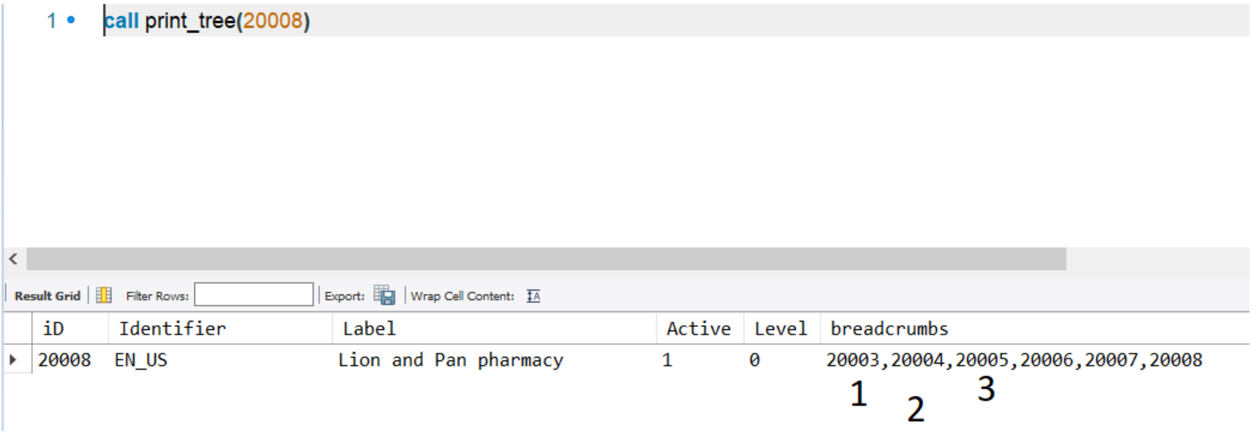
```sql
SELECT
* FROM concept
where Label like '%Lion and pan pharmacy%';
```



```
   ID     Identifier   Label                                        Active
▸  20008  EN_US        Lion and Pan pharmacy                        1
   20009  ne_NP        Lion and Pan pharmacy                        0
*  NULL   NULL         NULL                                         NULL
```

Figure 21. Search by the name

Then, will explore any concept above, using the “print_tree”, e.g. 20008



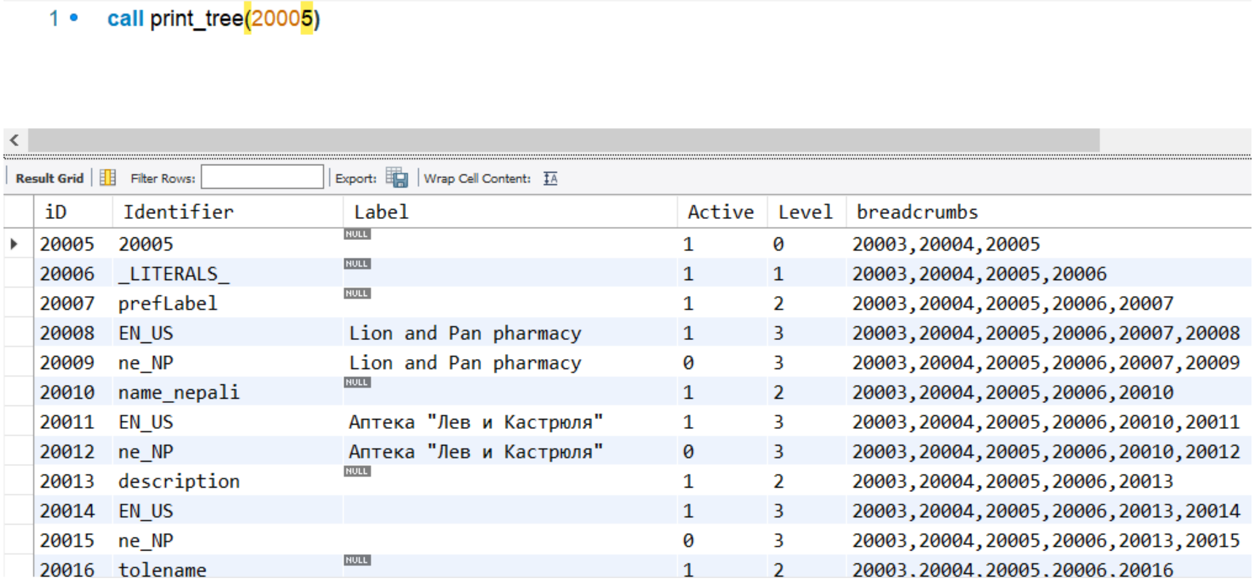
```
call print_tree(20008)
```

iD	Identifier	Label	Active	Level	breadcrumbs
20008	EN_US	Lion and Pan pharmacy	1	0	20003,20004,20005,20006,20007,20008

Figure 22. Search for the root concept of the application data

1. The common root for all applications of this type is available in the “tree_root” view.
2. The email of the applicant.
3. The root concept of the application

Then, will explore the application found. The root concept of it is 20005.



```
call print_tree(20005)
```

iD	Identifier	Label	Active	Level	breadcrumbs
20005	20005	NULL	1	0	20003,20004,20005
20006	_LITERALS_	NULL	1	1	20003,20004,20005,20006
20007	prefLabel	NULL	1	2	20003,20004,20005,20006,20007
20008	EN_US	Lion and Pan pharmacy	1	3	20003,20004,20005,20006,20007,20008
20009	ne_NP	Lion and Pan pharmacy	0	3	20003,20004,20005,20006,20007,20009
20010	name_nepali	NULL	1	2	20003,20004,20005,20006,20010
20011	EN_US	Аптека "Лев и Кастрюля"	1	3	20003,20004,20005,20006,20010,20011
20012	ne_NP	Аптека "Лев и Кастрюля"	0	3	20003,20004,20005,20006,20010,20012
20013	description	NULL	1	2	20003,20004,20005,20006,20013
20014	EN_US		1	3	20003,20004,20005,20006,20013,20014
20015	ne_NP		0	3	20003,20004,20005,20006,20013,20015
20016	tolename	NULL	1	2	20003,20004,20005,20006,20016

Figure 23. The application data example

The Pharmadex 2 software uses the ORM object Thing to assembly the application data. The whole ORM structure is below (Figure 24).

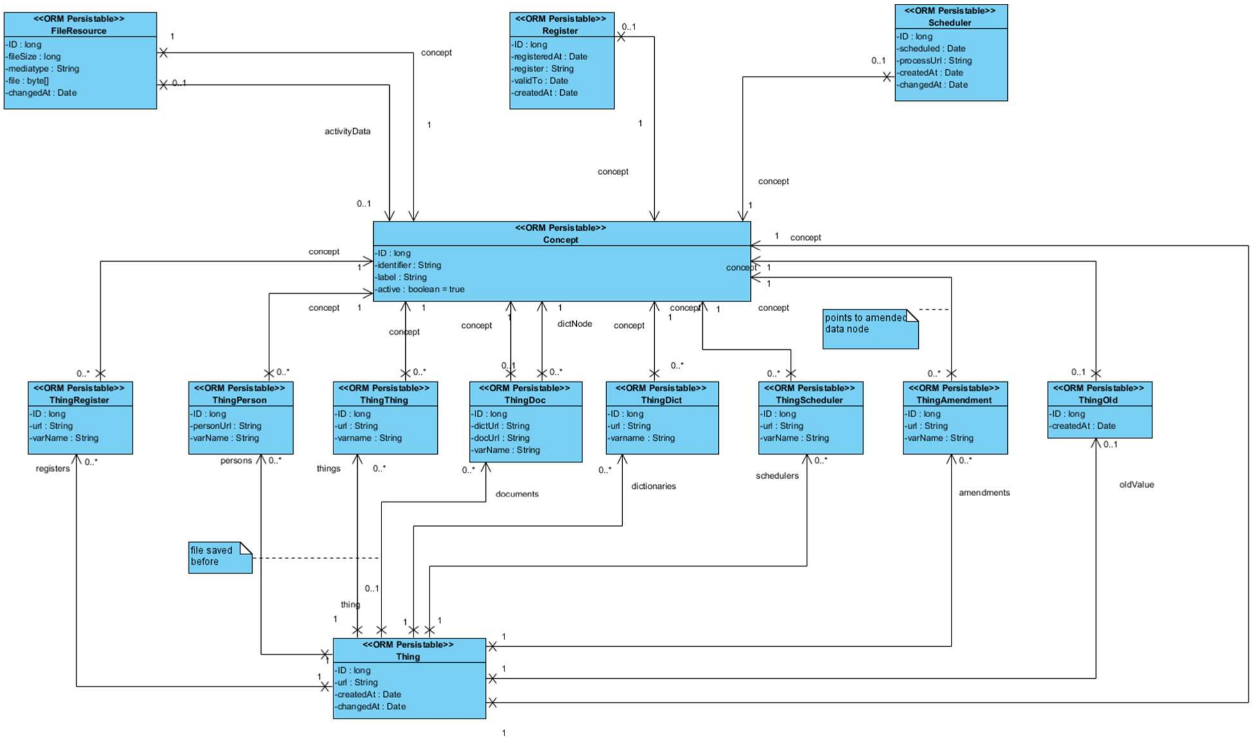


Figure 24. Full application data structure

The part of the whole ORM structure (Figure 24) are relations between the application data concept and files attached to it (Figure 25).

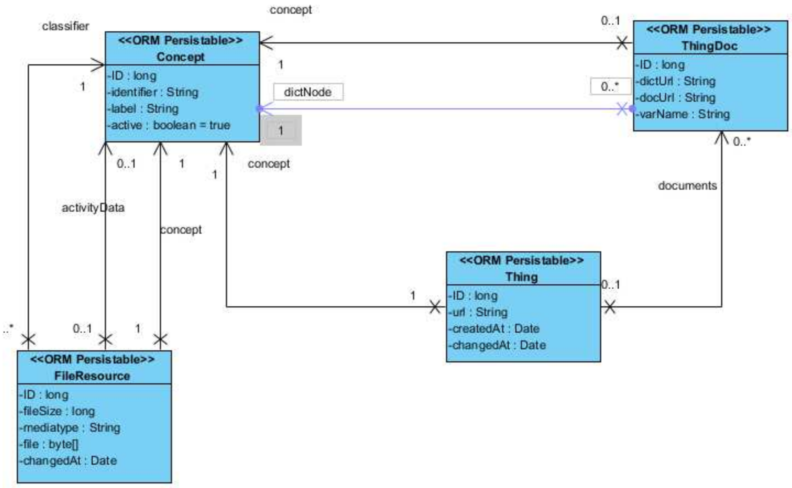
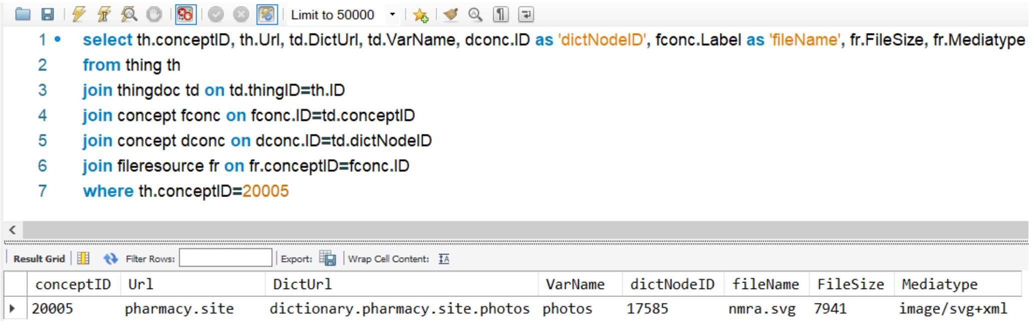


Figure 25. Files attached to the application data

- The application data concept is related to the Thing
- The Thing is related to a zero or many objects ThingDoc
- The ThingDoc is related to a file concept object and a concept object in a dictionary
- The file concept object is related to a file stored

To explain the considerations above, it is possible to create a SQL to extract all uploaded files related to the application data concept with ID=20005.



```sql
select th.conceptID, th.Url, td.DictUrl, td.VarName, dconc.ID as 'dictNodeID', fconc.Label as 'fileName', fr.FileSize, fr.Mediatype
from thing th
join thingdoc td on td.thingID=th.ID
join concept fconc on fconc.ID=td.conceptID
join concept dconc on dconc.ID=td.dictNodeID
join fileresource fr on fr.conceptID=fconc.ID
where th.conceptID=20005
```

```
conceptID  Url            DictUrl                          VarName  dictNodeID  fileName  FileSize  Mediatype
20005      pharmacy.site  dictionary.pharmacy.site.photos  photos   17585       nmra.svg  7941      image/svg+xml
```

Figure 26. Uploaded file in the application data. SQL

QUIZ

1) The ThingDoc object relates to concept dictNode. It is a concept of a dictionary item. Why do we need this reference?
   a) To provide a user a guide on which file should be uploaded
   b) To check file name against a dictionary item
   c) To attach the file to the dictionary
2) The query listed in Figure 26 returns more than one record. Is it possible?
   a) Yes
   b) No
3) The query listed in Figure 26 returns zero records. Is it possible?
   a) Yes
   b) No

MODULE 7. THE APPLICATION DATA AND ADDITIONAL DATA

OBJECTIVE

To learn how to create a sequency of electronic forms to prepare an application. The pre-defined data components and the additional data.

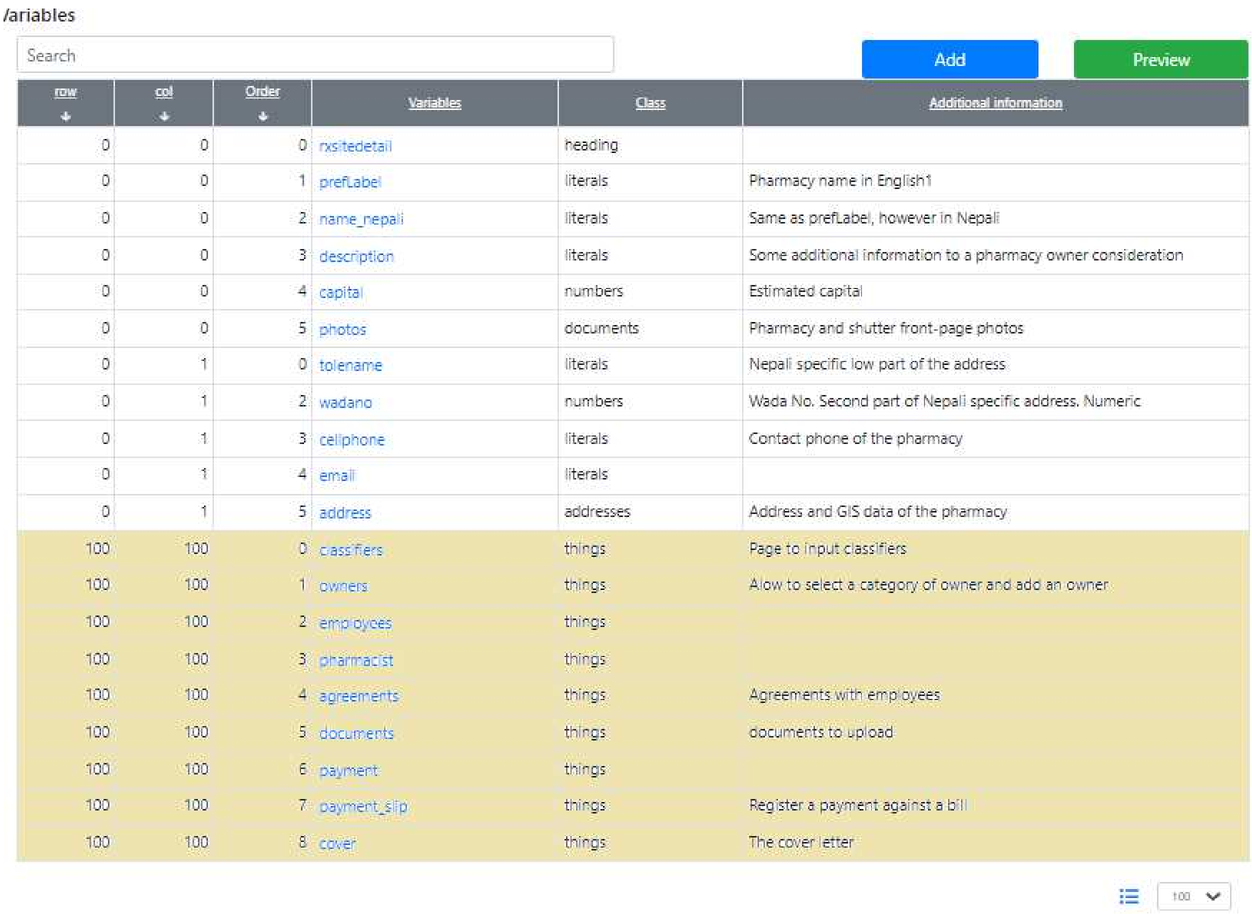
SUMMARY

A typical application is a sequency of electronic forms. This sequency is configurable by the Supervisor user. An electronic form consists of pre-defined data elements. The examples of them are:

- data input field
- dictionary to select value(s)
- address to select an address and geo-data
- file uploader
- file downloader
- etc.

The pre-defined data elements are highly customizable by the Supervisor user.

The first on-screen form in an application is the application data. The others on-screen forms are additional data. Altogether is the application. Below (Figure 27), the additional data forms are highlighted, pre-defined data components are on white background.



row	col	Order	Variables	Class	Additional information
0	0	0	rxsitedetail	heading	
0	0	1	prefLabel	literals	Pharmacy name in English1
0	0	2	name_nepali	literals	Same as prefLabel, however in Nepali
0	0	3	description	literals	Some additional information to a pharmacy owner consideration
0	0	4	capital	numbers	Estimated capital
0	0	5	photos	documents	Pharmacy and shutter front-page photos
0	1	0	tolename	literals	Nepali specific low part of the address
0	1	2	wadano	numbers	Wada No. Second part of Nepali specific address. Numeric
0	1	3	cellphone	literals	Contact phone of the pharmacy
0	1	4	email	literals	
0	1	5	address	addresses	Address and GIS data of the pharmacy
100	100	0	classifiers	things	Page to input classifiers
100	100	1	owners	things	Alow to select a category of owner and add an owner
100	100	2	employees	things	
100	100	3	pharmacist	things	
100	100	4	agreements	things	Agreements with employees
100	100	5	documents	things	documents to upload
100	100	6	payment	things	
100	100	7	payment_slip	things	Register a payment against a bill
100	100	8	cover	things	The cover letter

Figure 27. An application with application data and additional data

The database implementation of the additional data is on the following diagram:

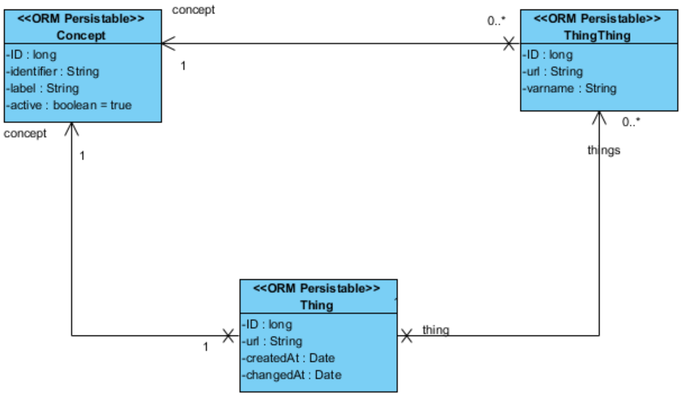
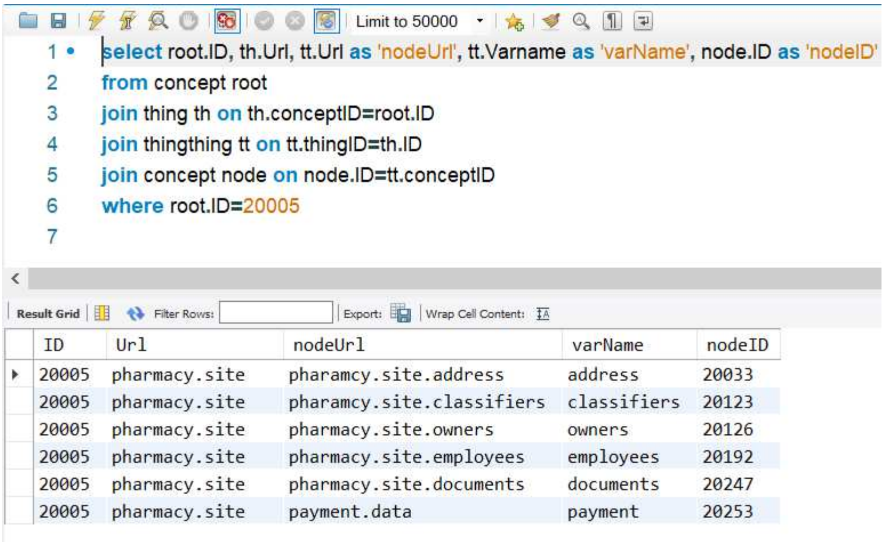


Figure 28. Representation of the application data and additional data in the database

- An application data concept object is related to a Thing object
- A thing object is related to zero or many ThingThing objects
- A ThingThing object is related to an additional data concept

To explain the considerations above, it is possible to create a SQL to extract all additional concepts for the application data concept (ID=20005).

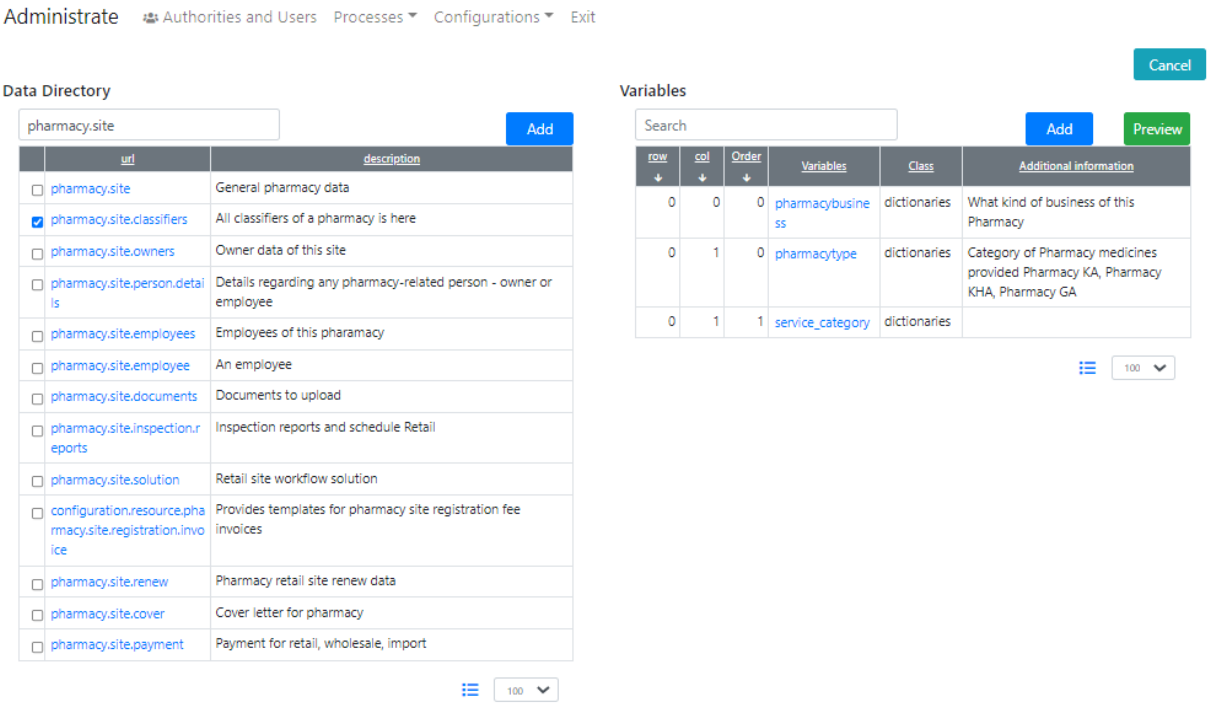


```
select root.ID, th.Url, tt.Url as 'nodeUrl', tt.Varname as 'varName', node.ID as 'nodeID'
from concept root
join thing th on th.conceptID=root.ID
join thingthing tt on tt.thingID=th.ID
join concept node on node.ID=tt.conceptID
where root.ID=20005
```

ID	Url	nodeUrl	varName	nodeID
20005	pharmacy.site	pharamcy.site.address	address	20033
20005	pharmacy.site	pharamcy.site.classifiers	classifiers	20123
20005	pharmacy.site	pharmacy.site.owners	owners	20126
20005	pharmacy.site	pharmacy.site.employees	employees	20192
20005	pharmacy.site	pharmacy.site.documents	documents	20247
20005	pharmacy.site	payment.data	payment	20253

Figure 29. Additional concepts to the application data

The additional data should be configured separately. An additional data should be configured only using data components. Example is “pharmacy.site.classifiers”. This additional data form consists of three dictionaries.



Data Directory

pharmacy.site

	url	description
□	pharmacy.site	General pharmacy data
☑	pharmacy.site.classifiers	All classifiers of a pharmacy is here
□	pharmacy.site.owners	Owner data of this site
□	pharmacy.site.person.details	Details regarding any pharmacy-related person - owner or employee
□	pharmacy.site.employees	Employees of this pharamacy
□	pharmacy.site.employee	An employee
□	pharmacy.site.documents	Documents to upload
□	pharmacy.site.inspection.reports	Inspection reports and schedule Retail
□	pharmacy.site.solution	Retail site workflow solution
□	configuration.resource.pharmacy.site.registration.invoice	Provides templates for pharmacy site registration fee invoices
□	pharmacy.site.renew	Pharmacy retail site renew data
□	pharmacy.site.cover	Cover letter for pharmacy
□	pharmacy.site.payment	Payment for retail, wholesale, import

Variables

row	col	Order	Variables	Class	Additional information
0	0	0	pharmacybusiness	dictionaries	What kind of business of this Pharmacy
0	1	0	pharmacytype	dictionaries	Category of Pharmacy medicines provided Pharmacy KA, Pharmacy KHA, Pharmacy GA
0	1	1	service_category	dictionaries	

Figure 30. Additional data configuration

This additional data may be used in any application data. For example, the same pharmacy site classifiers may suit different new pharmacy site applications.

QUIZ

1) Is it possible that the application data fits in one on-screen form?
2) The application data consists of the application data and sixteen additional data forms.
    a) Is it possible technically?
        i) Yes
        ii) No
    b) Is it acceptable?
        i) Yes
        ii) No
3) Who is solely responsible for application data configuration?
    a) Supervisor
    b) Moderator
    c) Programmer
    d) Business User
4) Can I use the same “pharmacy.site.employees” additional data configuration for retail and wholesale pharmacies?
    a) Yes
    b) No

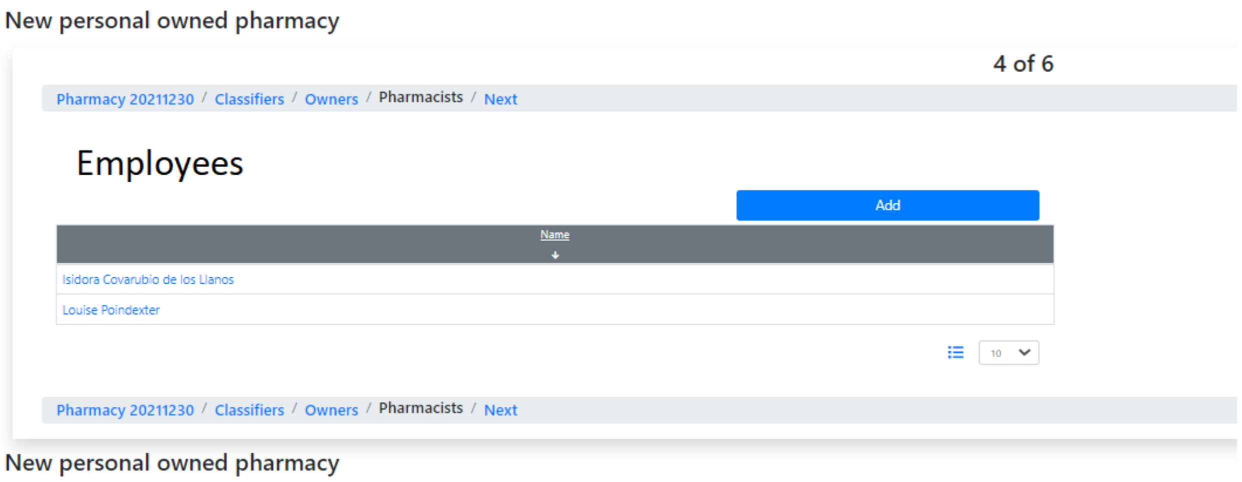
Module 8. Manage persons and ingredients

Objective

How to manage a list of persons or ingredients attached to an additional data

Summary

The pre-defined data components “persons” and “ingredients” provide a list of persons or medicine ingredients. These components should be placed to the additional data. An example is a list of pharmacy employees (Figure 31).



New personal owned pharmacy

4 of 6

Pharmacy 20211230 / Classifiers / Owners / Pharmacists / Next

Employees

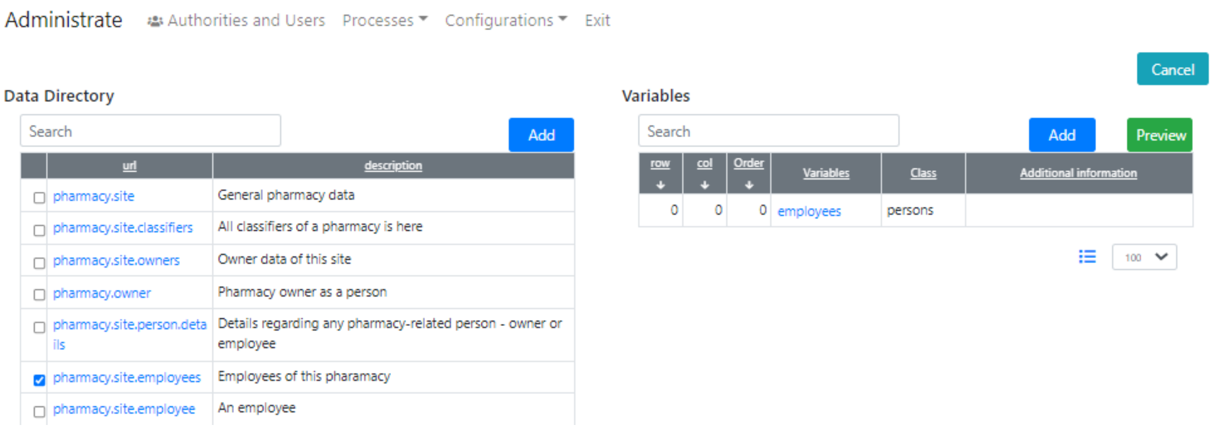
Name
Isidora Covarubio de los Llanos
Louise Poindexter

Pharmacy 20211230 / Classifiers / Owners / Pharmacists / Next

New personal owned pharmacy

Figure 31. The "persons" pred-defined data component in the user interface

The list above (Figure 31) is implemented by the “persons” data component included in the “pharmacy.site.employees” data configuration (Figure 32).



Data Directory

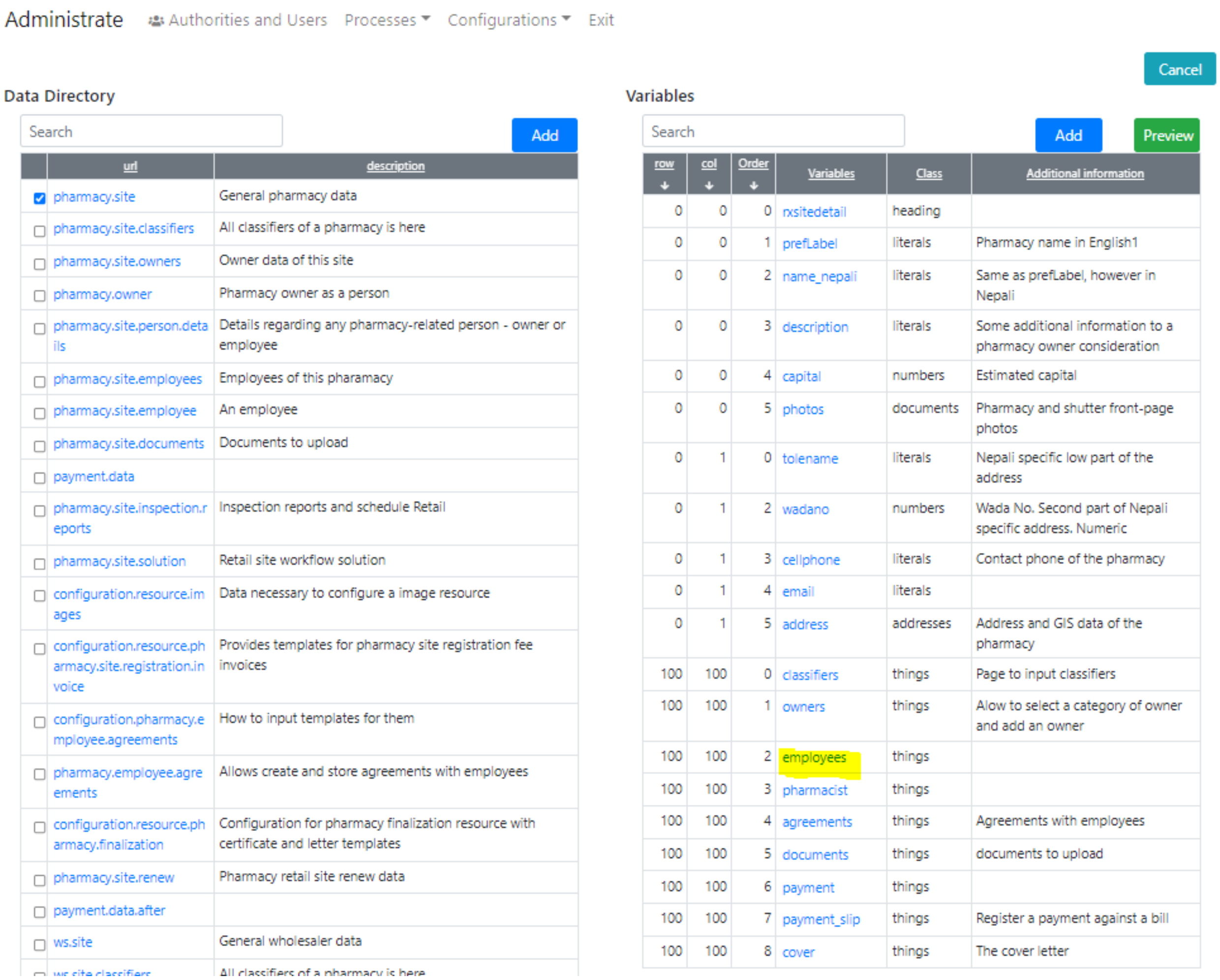
	url	description
□	pharmacy.site	General pharmacy data
□	pharmacy.site.classifiers	All classifiers of a pharmacy is here
□	pharmacy.site.owners	Owner data of this site
□	pharmacy.owner	Pharmacy owner as a person
□	pharmacy.site.person.details	Details regarding any pharmacy-related person - owner or employee
☑	pharmacy.site.employees	Employees of this pharamacy
□	pharmacy.site.employee	An employee

Variables

row	col	Order	Variables	Class	Additional information
0	0	0	employees	persons	

Figure 32. The pre-defined "persons" data component the additional data configuration

In its turn, this data configuration is the additional data configuration in the "pharmacy.site" application data (Figure 33).

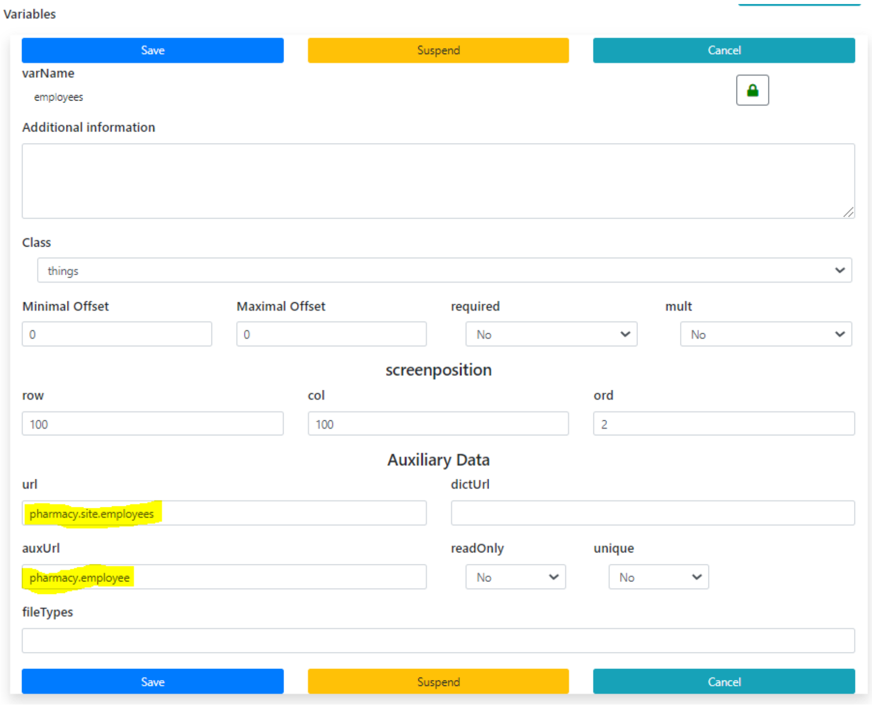


	url	description
☑	pharmacy.site	General pharmacy data
□	pharmacy.site.classifiers	All classifiers of a pharmacy is here
□	pharmacy.site.owners	Owner data of this site
□	pharmacy.owner	Pharmacy owner as a person
□	pharmacy.site.person.details	Details regarding any pharmacy-related person - owner or employee
□	pharmacy.site.employees	Employees of this pharamacy
□	pharmacy.site.employee	An employee
□	pharmacy.site.documents	Documents to upload
□	payment.data	
□	pharmacy.site.inspection.reports	Inspection reports and schedule Retail
□	pharmacy.site.solution	Retail site workflow solution
□	configuration.resource.images	Data necessary to configure a image resource
□	configuration.resource.pharmacy.site.registration.invoice	Provides templates for pharmacy site registration fee invoices
□	configuration.pharmacy.employee.agreements	How to input templates for them
□	pharmacy.employee.agreements	Allows create and store agreements with employees
□	configuration.resource.pharmacy.finalization	Configuration for pharmacy finalization resource with certificate and letter templates
□	pharmacy.site.renew	Pharmacy retail site renew data
□	payment.data.after	
□	ws.site	General wholesaler data
□	ws.site.classifiers	All classifiers of a pharmacy is here

row ↓	col ↓	Order ↓	Variables	Class	Additional information
0	0	0	rxsitedetail	heading	
0	0	1	prefLabel	literals	Pharmacy name in English1
0	0	2	name_nepali	literals	Same as prefLabel, however in Nepali
0	0	3	description	literals	Some additional information to a pharmacy owner consideration
0	0	4	capital	numbers	Estimated capital
0	0	5	photos	documents	Pharmacy and shutter front-page photos
0	1	0	tolename	literals	Nepali specific low part of the address
0	1	2	wadano	numbers	Wada No. Second part of Nepali specific address. Numeric
0	1	3	cellphone	literals	Contact phone of the pharmacy
0	1	4	email	literals	
0	1	5	address	addresses	Address and GIS data of the pharmacy
100	100	0	classifiers	things	Page to input classifiers
100	100	1	owners	things	Alow to select a category of owner and add an owner
100	100	2	employees	things	
100	100	3	pharmacist	things	
100	100	4	agreements	things	Agreements with employees
100	100	5	documents	things	documents to upload
100	100	6	payment	things	
100	100	7	payment_slip	things	Register a payment against a bill
100	100	8	cover	things	The cover letter

Figure 33. The list of persons is included as the additional data

The configuration parameter auxURL is required for the "persons" pred-defined data component. This parameter should be defined in the additional data "employees" of "pharmacy.site" configuration



varName

employees

Additional information

Class

things

Minimal Offset	Maximal Offset	required	mult
0	0	No	No

screenposition

row	col	ord
100	100	2

Auxiliary Data

url: pharmacy.site.employees

dictUrl:

auxUrl: pharmacy.employee

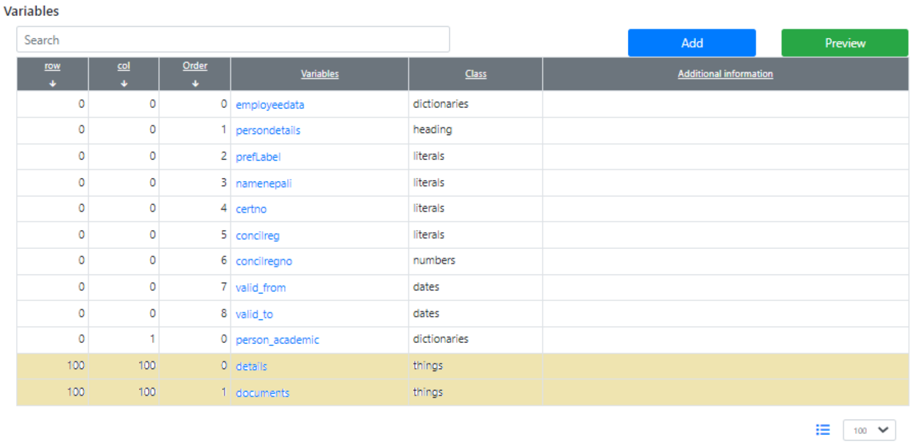
readOnly: No

unique: No

fileTypes

Figure 34. The auxUrl field points to the data configuration for an employee person – "pharmacy.employee"

The "pharmacy.employee" may include additional data.



row	col	Order	Variables	Class	Additional information
0	0	0	employeedata	dictionaries	
0	0	1	persondetails	heading	
0	0	2	prefLabel	literals	
0	0	3	namenepali	literals	
0	0	4	certno	literals	
0	0	5	concilreg	literals	
0	0	6	concilregno	numbers	
0	0	7	valid_from	dates	
0	0	8	valid_to	dates	
0	1	0	person_academic	dictionaries	
100	100	0	details	things	
100	100	1	documents	things	

Figure 35. The additional data in "pharmacy.site.employee"

The auxURL parameter is to define the configuration and storage for one element of the list of employees (Figure 36, Figure 37).

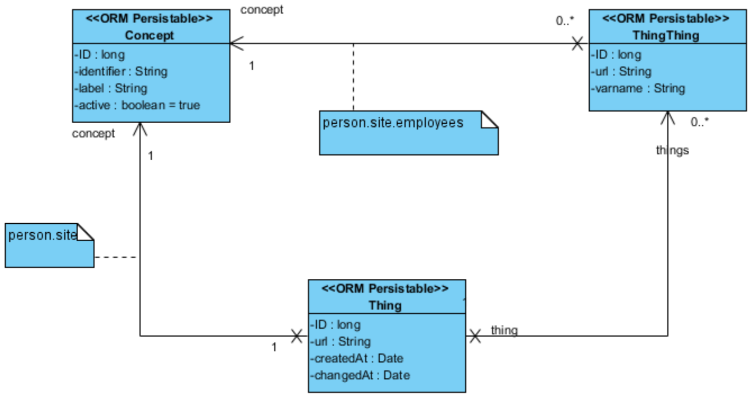


Figure 36. A concept "person.site.employees" is an additional data for application data "person.site"

A concept “person.site.employees” is linked to a concept “pharmacy.site.employee” using Thing Person.

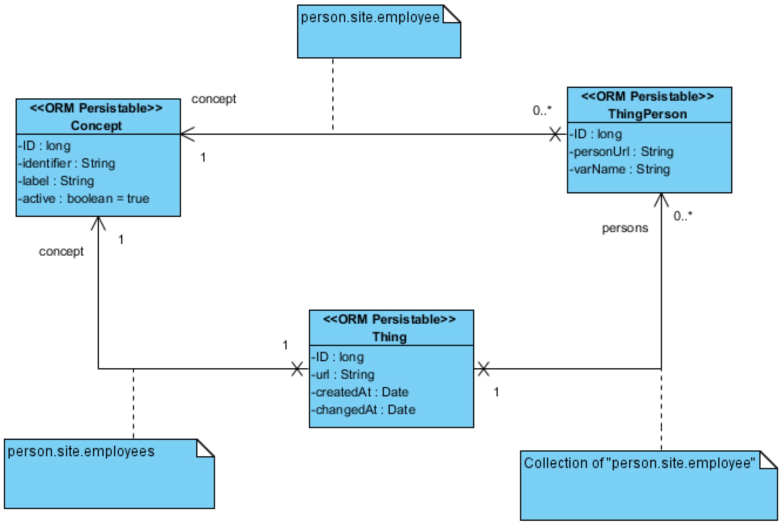
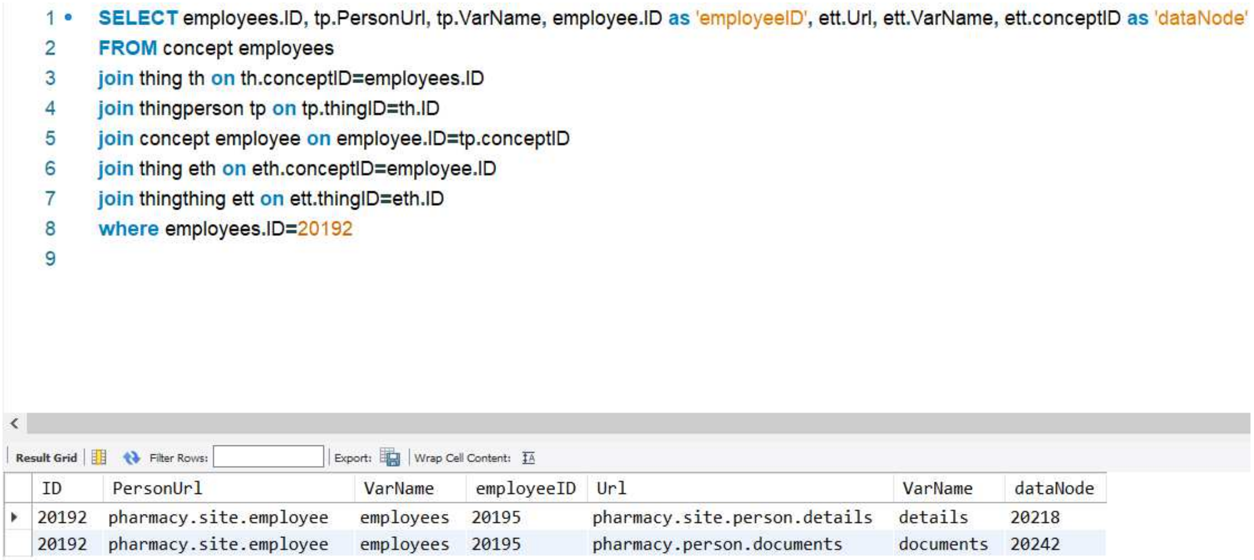


Figure 37. An application node "pharmacy.site.employees" is linked to zero or many nodes "pharmacy.site.employee"

To explain the considerations above, it is possible to create a SQL to extract all additional data concepts from “pharmacy.site.employee” linked to the “pharmacy.site.employees” concept with ID=20192 (Figure 36)



```sql
SELECT employees.ID, tp.PersonUrl, tp.VarName, employee.ID as 'employeeID', ett.Url, ett.VarName, ett.conceptID as 'dataNode'
FROM concept employees
join thing th on th.conceptID=employees.ID
join thingperson tp on tp.thingID=th.ID
join concept employee on employee.ID=tp.conceptID
join thing eth on eth.conceptID=employee.ID
join thingthing ett on ett.thingID=eth.ID
where employees.ID=20192
```

```
ID     PersonUrl               VarName    employeeID  Url                          VarName    dataNode
20192  pharmacy.site.employee  employees  20195       pharmacy.site.person.details  details    20218
20192  pharmacy.site.employee  employees  20195       pharmacy.person.documents     documents  20242
```

Figure 38. SQL to extract "pharmacy.site.employees" data

QUIZ

1) Is it possible to place a list of product manufacturers?
   a) Yes
   b) No
2) 7) Is it possible to place a text field to the on-screen form that contains the "persons" pre-defined data component?
   a) Yes
   b) No
3) Is it possible to use the pre-defined component "ingredients" in the additional data for medical devices?
   a) Yes
   b) No
4) A pre-defined component consists of User Interface and server-side API. Does the "persons" component API use the ThingPerson ORM class in the server-side codes?
   a) Yes
   b) No
5) Can SQL query on Figure 38 return data for all employees
   a) Yes
   b) No

Module 9. The workflow data

Objective

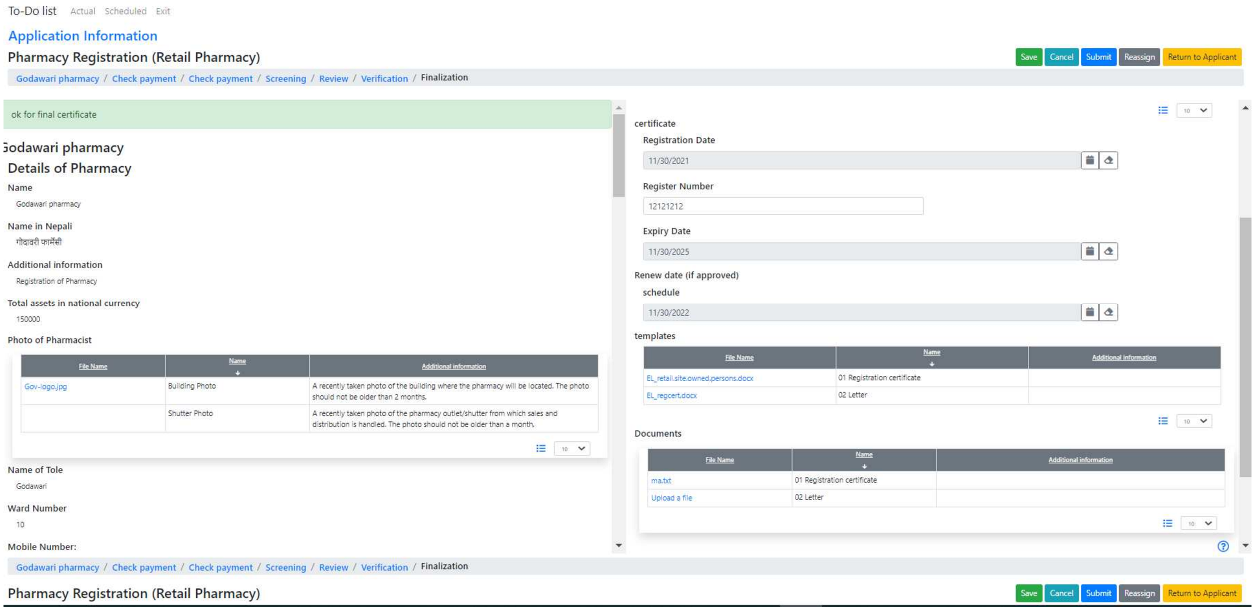
To study how to get data collected while application processing in NMRA.

Summary

In addition to the application data defined by the applicant, the Pharmadex 2 collects application-related data defined by NMRA experts. This data is a result of workflow activities execution, i.e., workflow data. Examples are:

- Certificates
- Register numbers
- Review, inspection, and laboratory reports
- Payment amounts
- Additional documents
- etc.

On the left side of Figure 39 is the workflow data



Application Information

Pharmacy Registration (Retail Pharmacy)

Godawari pharmacy / Check payment / Check payment / Screening / Review / Verification / Finalization

ok for final certificate

Godawari pharmacy

Details of Pharmacy

Name
Godawari pharmacy

Name in Nepali
गोदावरी फार्मेसी

Additional information
Registration of Pharmacy

Total assets in national currency
150000

Photo of Pharmacist

File Name	Name	Additional information
Gov-logo.jpg	Building Photo	A recently taken photo of the building where the pharmacy will be located. The photo should not be older than 2 months.
	Shutter Photo	A recently taken photo of the pharmacy outlet/shutter from which sales and distribution is handled. The photo should not be older than a month.

Name of Tole
Godawari

Ward Number
10

Mobile Number:

certificate

Registration Date
11/30/2021

Register Number
12121212

Expiry Date
11/30/2025

Renew date (if approved)

schedule
11/30/2022

templates

File Name	Name	Additional information
EL_retail.site.owned.persons.docx	01 Registration certificate	
EL_regcert.docx	02 Letter	

Documents

File Name	Name	Additional information
ma.txt	01 Registration certificate	
Upload a file	02 Letter	

Godawari pharmacy / Check payment / Check payment / Screening / Review / Verification / Finalization

Pharmacy Registration (Retail Pharmacy)

Figure 39. The workflow data in the User Interface

The workflow data and application data relation are implemented by the ORM object “History” (Figure 40)

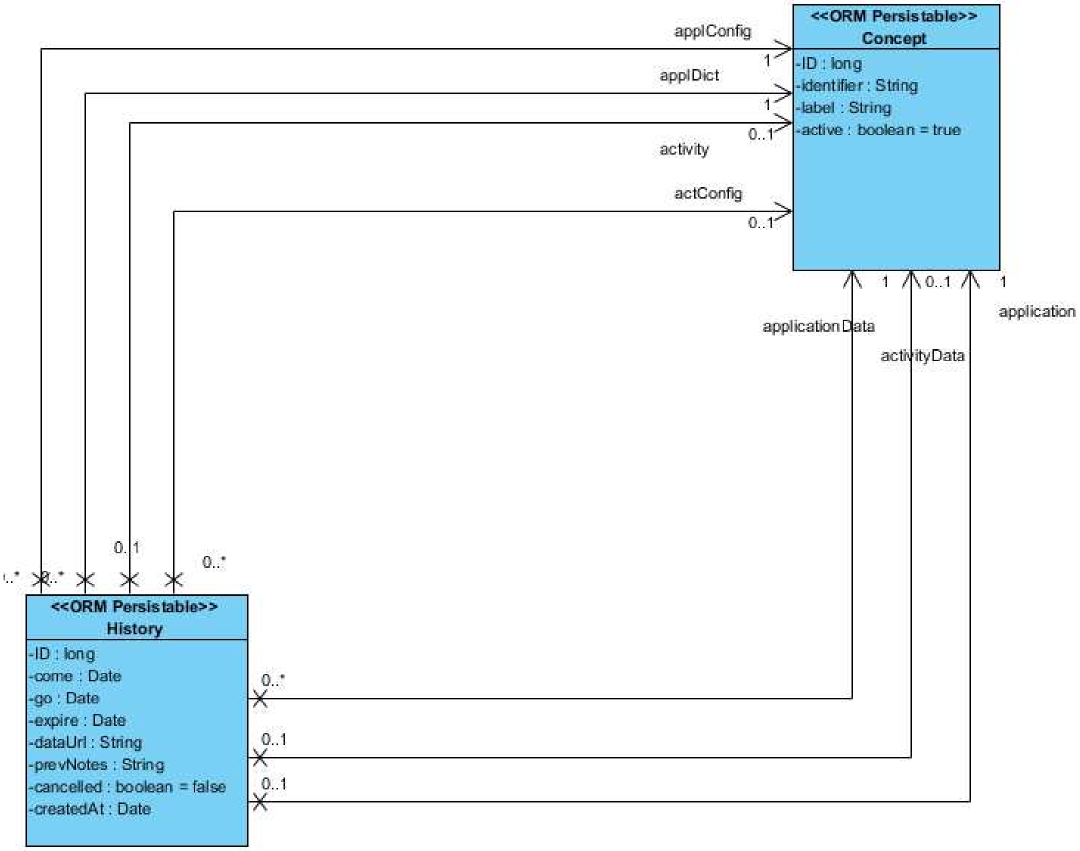


Figure 40. The History object is a hub for concepts

These relations are:

- applConfig – a concept to manage a configuration of the workflow
- applDict – a concept that belongs to dictionary item that keeps name and description of a workflow
- activity – a concept to manage the activity of the workflow
- actConfig – a concept to manage a configuration of the activity of the workflow.
- applicationData – a concept to manage an application data
- activityData – a concept to manage an activity data
- application – a concept to manage the whole application

The Pharmadex 2 stores the workflow data the same way as the application data – under root URL and Owner concepts. Suppose the root URL is “pharmacy.site.renew” (Figure 41)

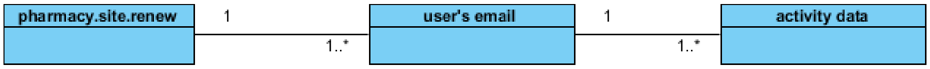
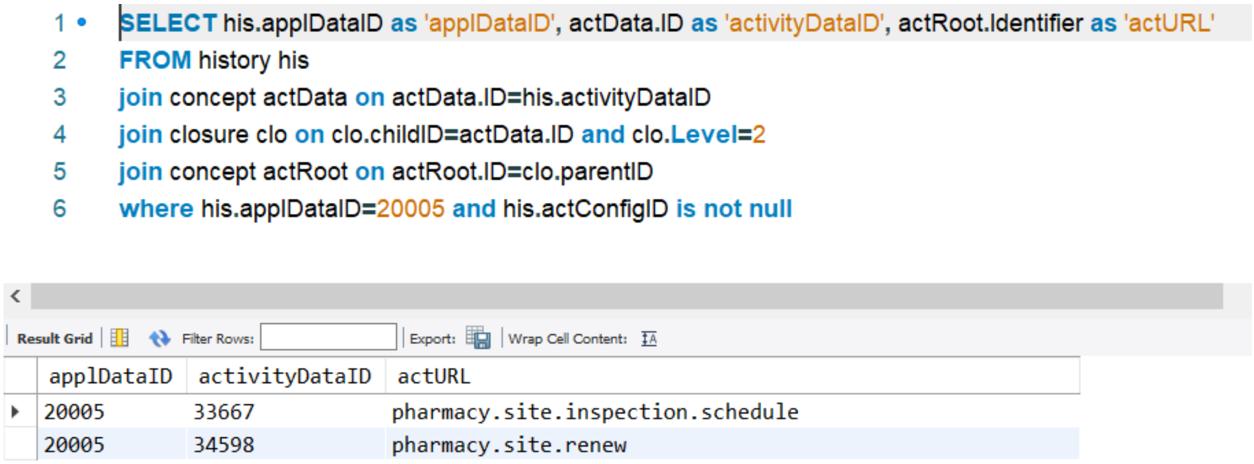


Figure 41. The workflow data is stored in the same structure as the application data

It is possible to use SQL to get all workflow data for application data with ID=20005 (Figure 42). Line 4 allows getting the root concept of the application data. The criteria “his.actConfigID is not null” allows selecting only workflow activities.



```
1  SELECT his.applDataID as 'applDataID', actData.ID as 'activityDataID', actRoot.Identifier as 'actURL'
2  FROM history his
3  join concept actData on actData.ID=his.activityDataID
4  join closure clo on clo.childID=actData.ID and clo.Level=2
5  join concept actRoot on actRoot.ID=clo.parentID
6  where his.applDataID=20005 and his.actConfigID is not null
```

applDataID	activityDataID	actURL
20005	33667	pharmacy.site.inspection.schedule
20005	34598	pharmacy.site.renew

Figure 42. Get all workflow data for application data ID=20005

QUIZ

1) Can be INN in the workflow data?
    a) Yes
    b) No
2) Line 4 in SQL in Figure 42 contains criteria “clo.Level=2”. What will display in column actURL in case “clo.Level=1”?
    a) error
    b) owner’s eMail
    c) The ID of a concept
3) Line 6 contains the criteria “his.actConfigID is not null”. What does it mean?
    a) Some history records may not point to an activity of the workflow.
    b) Some workflow activities haven't configurations

MODULE 10. REPORTS, OR PUT IT ALL TOGETHER

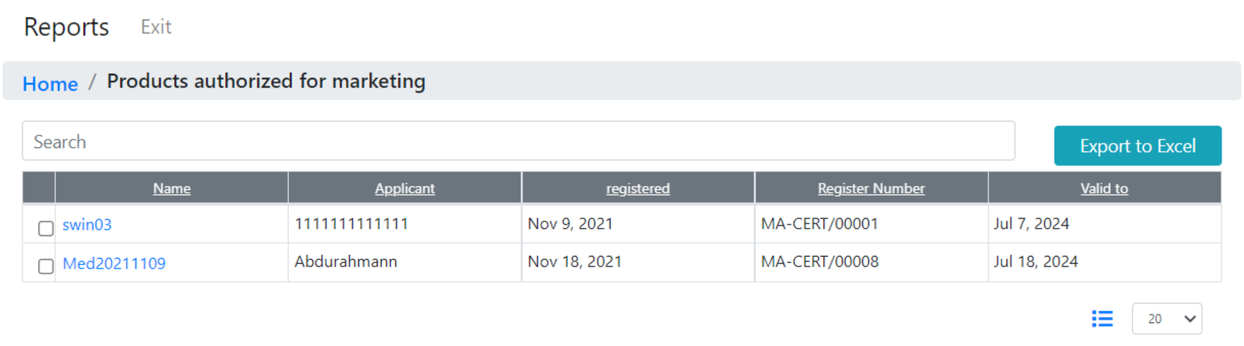
OBJECTIVE

The learning Pharmadex 2 basic report feature will help to get a diverse knowledge of how to query the data.

SUMMARY

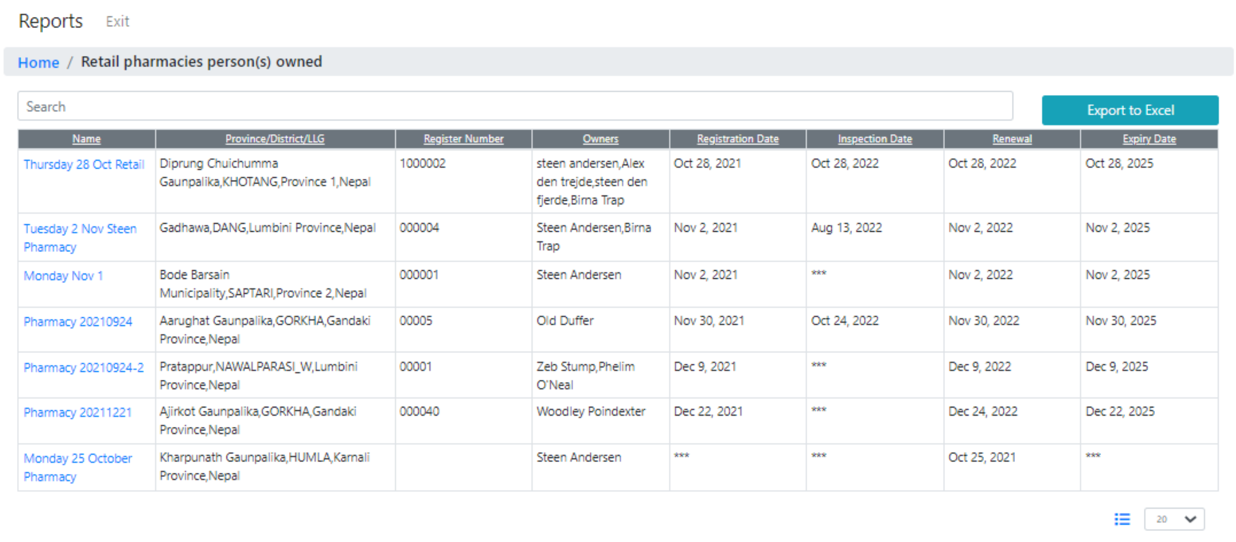
The basic report feature of Pharmadex 2 allows the reporting of medical products and medical sites. The medical products are medicines and medical devices. The medical sites are pharmacies, warehouses, manufactures, etc.

The reports look like Figure 43 and Figure 44



	Name	Applicant	registered	Register Number	Valid to
□	swin03	1111111111111	Nov 9, 2021	MA-CERT/00001	Jul 7, 2024
□	Med20211109	Abdurahmann	Nov 18, 2021	MA-CERT/00008	Jul 18, 2024

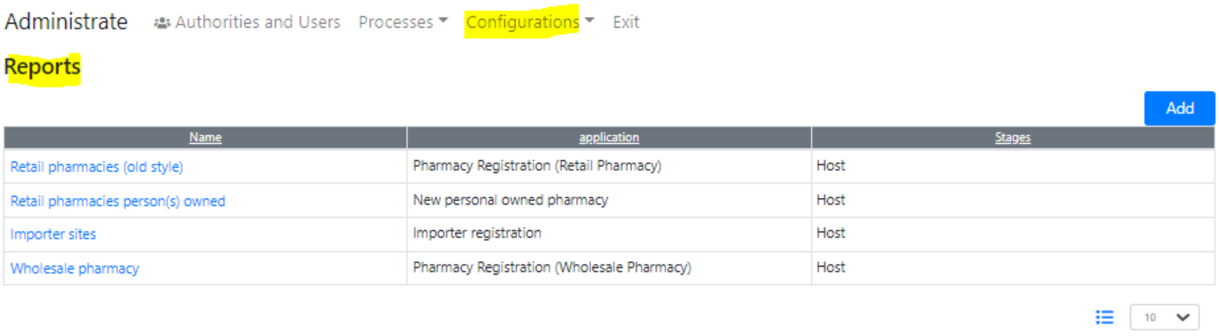
Figure 43. Medical products report



Name	Province/District/LLG	Register Number	Owners	Registration Date	Inspection Date	Renewal	Expiry Date
Thursday 28 Oct Retail	Diprung Chuichumma Gaunpalika,KHOTANG,Province 1,Nepal	1000002	steen andersen,Alex den trejde,steen den fjerde,Bima Trap	Oct 28, 2021	Oct 28, 2022	Oct 28, 2022	Oct 28, 2025
Tuesday 2 Nov Steen Pharmacy	Gadhawa,DANG,Lumbini Province,Nepal	000004	Steen Andersen,Birna Trap	Nov 2, 2021	Aug 13, 2022	Nov 2, 2022	Nov 2, 2025
Monday Nov 1	Bode Barsain Municipality,SAPTARI,Province 2,Nepal	000001	Steen Andersen	Nov 2, 2021	***	Nov 2, 2022	Nov 2, 2025
Pharmacy 20210924	Aarughat Gaunpalika,GORKHA,Gandaki Province,Nepal	00005	Old Duffer	Nov 30, 2021	Oct 24, 2022	Nov 30, 2022	Nov 30, 2025
Pharmacy 20210924-2	Pratappur,NAWALPARASI_W,Lumbini Province,Nepal	00001	Zeb Stump,Phelim O'Neal	Dec 9, 2021	***	Dec 9, 2022	Dec 9, 2025
Pharmacy 20211221	Ajirkot Gaunpalika,GORKHA,Gandaki Province,Nepal	000040	Woodley Poindexter	Dec 22, 2021	***	Dec 24, 2022	Dec 22, 2025
Monday 25 October Pharmacy	Kharpunath Gaunpalika,HUMLA,Karnali Province,Nepal		Steen Andersen	***	***	Oct 25, 2021	***

Figure 44. Medical sites report

The Supervisor should configure a report for any type of application. The report configuration feature is available in Administrate group of tasks



Name	application	Stages
Retail pharmacies (old style)	Pharmacy Registration (Retail Pharmacy)	Host
Retail pharmacies person(s) owned	New personal owned pharmacy	Host
Importer sites	Importer registration	Host
Wholesale pharmacy	Pharmacy Registration (Wholesale Pharmacy)	Host

Figure 45. Reports configuration feature

All configurations are stored under the root URL “report. configuration”. The description of this structure is for the Supervisor, thus beyond this course.

To query data for reports, Pharmadex 2 provides a set of stored procedures. These stored procedures are in the database “pdx2”. The final reports will be produced by:

- report_products
- report_sites

The reports are uniform. This uniformity is backed by the tree data model and parameters of the stored procedures.

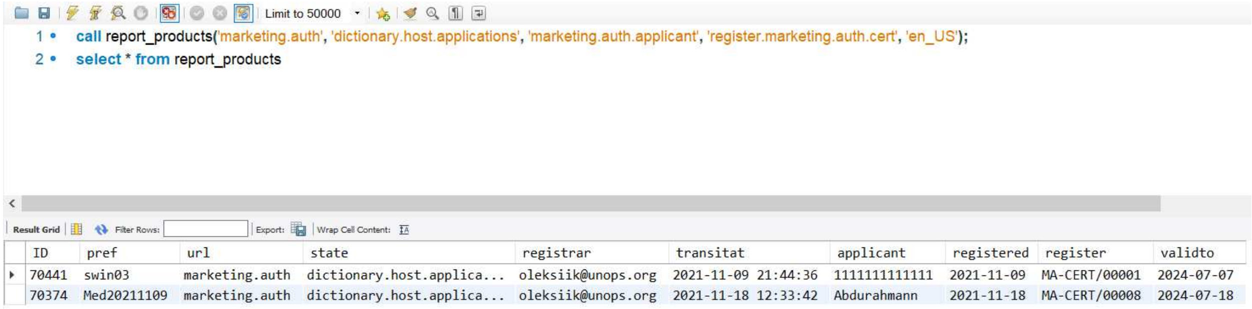
The Pharmadex 2 distinguishes product reports and the site reports by the “addressURL” parameter in the report configuration. The “addressURL” parameter is the site property.

The report_products procedure requires parameters:

Parameter	Description	Value example
data_url	Root URL of application data	marketing.auth
state_url	Any application data in any given moment is in some workflow. This workflow may be either started or scheduled to start. The workflow implements a lifecycle stage of the application data. Workflows for the same stage are listed in the same dictionary. The URL of this dictionary is a state URL.	The possible URLs are: • “dictionary.guest.applications” – new not approved • “dictionary.host.applications” – approved • dictionary.shutdown.applications – canceled or discharged, however existing in the database

Parameter	Description	Value example
applicant_url	URL under which applicant data has been stored. The applicant data may be configured by Supervisor. The text field “prefLabel” is mandatory[8]	marketing.auth.applicant
register_url	URL under which the registration number has been stored. This URL is particular for the application. This URL can be found in the appropriative workflow data configuration	register.marketing.auth.cert
lang	The language	en_US

An example is below



```
call report_products('marketing.auth', 'dictionary.host.applications', 'marketing.auth.applicant', 'register.marketing.auth.cert', 'en_US');
select * from report_products
```

```
  ID     pref         url             state                       registrar            transitat             applicant        registered   register        validto
▸ 70441  swin03       marketing.auth  dictionary.host.applica...  oleksiik@unops.org   2021-11-09 21:44:36   1111111111111    2021-11-09   MA-CERT/00001   2024-07-07
  70374  Med20211109  marketing.auth  dictionary.host.applica...  oleksiik@unops.org   2021-11-18 12:33:42   Abdurahmann      2021-11-18   MA-CERT/00008   2024-07-18
```

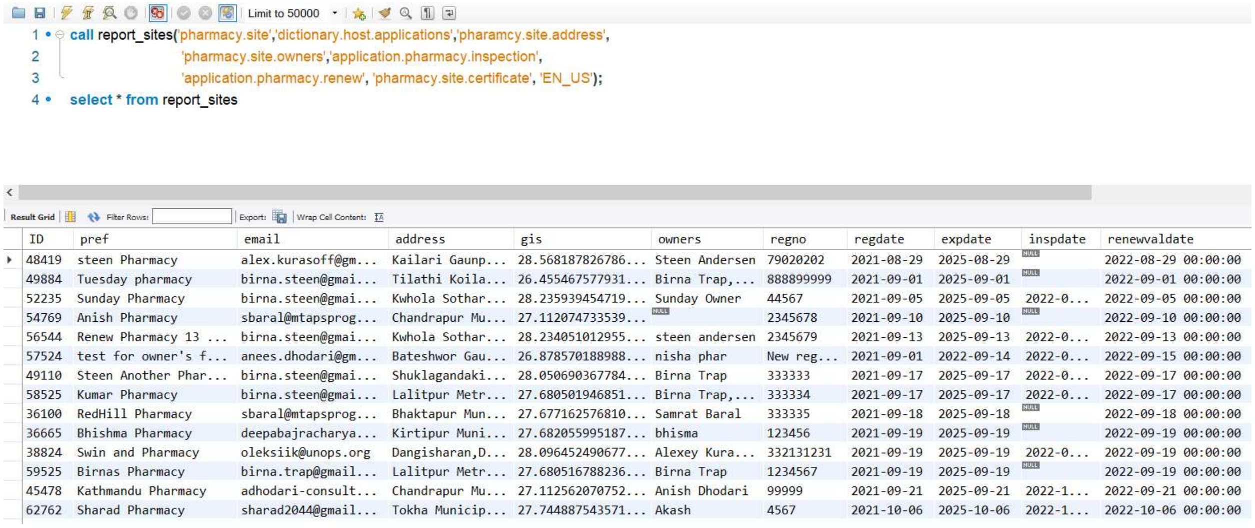
Figure 46. An example of report_products call

[8] The SKOS originated requirement. See https://www.w3.org/2012/09/odrl/semantic/draft/doco/skos_prefLabel.html

The report_sites procedure requires parameters:

Parameter	Description	Value example
site_url	Root URL of application data	pharmacy.site
dict_stage_url	Any application data in any given moment is in some workflow. This workflow may be either started or scheduled to start. The workflow implements a lifecycle stage of the application data. Workflows for the same stage are listed in the same dictionary. The URL of this dictionary is a state URL.	The possible URLs are: • "dictionary.guest.applications" – new not approved • "dictionary.host.applications" – approved • "dictionary.shutdown.applications" – canceled or discharged, however existing in the database
addr_url	The URL under which the address of a site is stored. Can be found in the data configuration.	pharamcy.site.address
owner_url	The URL under which owners of the site are stored. Can be found in the data configuration.	pharmacy.site.owners
appl_inspection_url	The URL under which the site inspection schedule is stored. This URL can be found in the appropriative workflow data configuration	application.pharmacy.inspection
appl_renew_url	The URL under which the site renewal schedule is stored. This URL can be found in the appropriative workflow data configuration	application.pharmacy.renew
appl_cert_url	The URL under which the site certificate register is stored. This URL can be found in the appropriative workflow data configuration	pharmacy.site.certificate
lang	The language	en_US

An example is below:



```
call report_sites('pharmacy.site','dictionary.host.applications','pharamcy.site.address',
                  'pharmacy.site.owners','application.pharmacy.inspection',
                  'application.pharmacy.renew', 'pharmacy.site.certificate', 'EN_US');
select * from report_sites
```

ID	pref	email	address	gis	owners	regno	regdate	expdate	inspdate	renewvaldate
48419	steen Pharmacy	alex.kurasoff@gm...	Kailari Gaunp...	28.568187826786...	Steen Andersen	79020202	2021-08-29	2025-08-29	NULL	2022-08-29 00:00:00
49884	Tuesday pharmacy	birna.steen@gmai...	Tilathi Koila...	26.455467577931...	Birna Trap,...	888899999	2021-09-01	2025-09-01	NULL	2022-09-01 00:00:00
52235	Sunday Pharmacy	birna.steen@gmai...	Kwhola Sothar...	28.235939454719...	Sunday Owner	44567	2021-09-05	2025-09-05	2022-0...	2022-09-05 00:00:00
54769	Anish Pharmacy	sbaral@mtapsprog...	Chandrapur Mu...	27.112074733539...	NULL	2345678	2021-09-10	2025-09-10	NULL	2022-09-10 00:00:00
56544	Renew Pharmacy 13 ...	birna.steen@gmai...	Kwhola Sothar...	28.234051012955...	steen andersen	2345679	2021-09-13	2025-09-13	2022-0...	2022-09-13 00:00:00
57524	test for owner's f...	anees.dhodari@gm...	Bateshwor Gau...	26.878570188988...	nisha phar	New reg...	2021-09-01	2022-09-14	2022-0...	2022-09-15 00:00:00
49110	Steen Another Phar...	birna.steen@gmai...	Shuklagandaki...	28.050690367784...	Birna Trap	333333	2021-09-17	2025-09-17	2022-0...	2022-09-17 00:00:00
58525	Kumar Pharmacy	birna.steen@gmai...	Lalitpur Metr...	27.680501946851...	Birna Trap,...	333334	2021-09-17	2025-09-17	2022-0...	2022-09-17 00:00:00
36100	RedHill Pharmacy	sbaral@mtapsprog...	Bhaktapur Mun...	27.677162576810...	Samrat Baral	333335	2021-09-18	2025-09-18	NULL	2022-09-18 00:00:00
36665	Bhishma Pharmacy	deepabajracharya...	Kirtipur Muni...	27.682055995187...	bhisma	123456	2021-09-19	2025-09-19	NULL	2022-09-19 00:00:00
38824	Swin and Pharmacy	oleksiik@unops.org	Dangisharan,D...	28.096452490677...	Alexey Kura...	332131231	2021-09-19	2025-09-19	2022-0...	2022-09-19 00:00:00
59525	Birnas Pharmacy	birna.trap@gmail...	Lalitpur Metr...	27.680516788236...	Birna Trap	1234567	2021-09-19	2025-09-19	NULL	2022-09-19 00:00:00
45478	Kathmandu Pharmacy	adhodari-consult...	Chandrapur Mu...	27.112562070752...	Anish Dhodari	99999	2021-09-21	2025-09-21	2022-1...	2022-09-21 00:00:00
62762	Sharad Pharmacy	sharad2044@gmail...	Tokha Municip...	27.744887543571...	Akash	4567	2021-10-06	2025-10-06	2022-1...	2022-10-06 00:00:00

Figure 47. An example of report_sites call

The stored procedures report_products and report_sites call others stored procedures and finally compose the result. These procedures are a good starting point to study the Pharmadex 2 data querying.

QUIZ

1) Is it mandatory to follow SKOS "prefLabel" recommendations to configure any application data?
   a) Yes
   b) No
2) Is it possible to get a list of all sorts of product applications using the "report_products" procedure?
   a) Yes
   b) No
3) Is it possible to create a uniform data query for all sorts of medical sites using the "report_sites" procedure?
   a) Yes
   b) No
4) May I change the existing stored procedures?
   a) Yes
   b) No

ANNEX I. SQL CODES

FIGURE 18. DATA CONFIGURATION SELECTION USE PLAIN SQL SELECT

```
select ct.*
from concept root
join closure clo on clo.parentID=root.ID
join concept ct on ct.ID=clo.childID
where root.ID=17513 and ct.Active
```

FIGURE 19. SELECT DATA CONFIGURATION USING SQL

```
select assm.`Row`, assm.Col, conf.Identifier, assm.clazz
from assembly assm
join (
select ct.*
from concept root
join closure clo on clo.parentID=root.ID
join concept ct on ct.ID=clo.childID
where root.ID=17513 and ct.Active
) conf on conf.ID=assm.conceptID
order by assm.`Row
```

FIGURE 13. SELECT A VALUE OF PREFLABEL IN EN_US

An original query:

```
select root.ID, var.Identifier as 'varName', pref.Identifier as 'language',pref.Label as 'prefLabel'
from concept root
join closure clo on clo.parentID=root.ID and clo.Level=1
join concept lit on lit.ID=clo.childID and lit.Identifier='_LITERALS_'
join closure clo1 on clo1.parentID=lit.ID and clo1.Level=1
join concept var on var.ID=clo1.childID and var.Identifier='prefLabel'
join closure clo2 on clo2.parentID=var.ID and clo2.Level=1
join concept pref on pref.ID=clo2.childID and pref.Identifier='EN_US'
where root.ID=20005
```

The simplest version of it:

```
select root.ID, var.Identifier as 'varName', pref.Identifier as 'language',pref.Label as 'prefLabel'
from concept root
join closure clo1 on clo1.parentID=root.ID and clo1.Level=2
join concept var on var.ID=clo1.childID and var.Identifier='prefLabel'
join closure clo2 on clo2.parentID=var.ID and clo2.Level=1
join concept pref on pref.ID=clo2.childID and pref.Identifier='EN_US'
where root.ID=20005
```

FIGURE 21. SEARCH BY THE NAME

```
SELECT
* FROM concept
```

```
where Label like '%Lion and pan pharmacy%';
```

Figure 26. Uploaded file in the application data. SQL

```
select th.conceptID, th.Url, td.DictUrl, td.VarName, dconc.ID as 'dictNodeID', fconc.Label as
'fileName', fr.FileSize, fr.Mediatype
from thing th
join thingdoc td on td.thingID=th.ID
join concept fconc on fconc.ID=td.conceptID
join concept dconc on dconc.ID=td.dictNodeID
join fileresource fr on fr.conceptID=fconc.ID
where th.conceptID=20005
```

Figure 29. Additional concepts to the application data

```
select root.ID, th.Url, tt.Url as 'nodeUrl', tt.Varname as 'varName', node.ID as 'nodeID'
from concept root
join thing th on th.conceptID=root.ID
join thingthing tt on tt.thingID=th.ID
join concept node on node.ID=tt.conceptID
where root.ID=20005
```

Figure 38. SQL to extract "pharmacy.site.employees" data

```
SELECT employees.ID, tp.PersonUrl, tp.VarName, employee.ID as 'employeeID', ett.Url,
ett.VarName, ett.conceptID as 'dataNode'
FROM concept employees
join thing th on th.conceptID=employees.ID
join thingperson tp on tp.thingID=th.ID
join concept employee on employee.ID=tp.conceptID
join thing eth on eth.conceptID=employee.ID
join thingthing ett on ett.thingID=eth.ID
where employees.ID=20192
```

Figure 42. Get all workflow data for application data ID=20005

```
SELECT his.applDataID as 'applDataID', actData.ID as 'activityDataID', actRoot.Identifier as
'actURL'
FROM history his
join concept actData on actData.ID=his.activityDataID
join closure clo on clo.childID=actData.ID and clo.Level=2
join concept actRoot on actRoot.ID=clo.parentID
where his.applDataID=20005 and his.actConfigID is not null
```

ANNEX 2. QUIZ ANSWERS

MODULE 1

1) The Pharmadex 2 software codes are open. What is the recommended way to adapt it to the needs of the particular NMRA?
   a) Re-programming some source codes
   b) Configuring Pharmadex 2. This answer is right, however only on the first stage of implementation.
   c) Configuring Pharmadex 2 and building particular APIs if it will be necessary. This answer is the most appropriate.
2) A dictionary in Pharmadex 2 is a data element represented as a graph of any reasonable depth. Which point of view is it?
   a) Data usage. The "graph" and "depth of graph" are from the logical model. For API development it is Concepts and Closures.
   b) API development
3) I'm a .NET programmer. How can I use project "pdxmodel"?
   a) As a software library
   b) For reference only
4) Is it possible to use Pharmadex 2 software API directly from the PHP application?
   a) Yes. Like any other JSON REST API
   b) No
5) We plan to implement a Microsoft Azure data warehouse. What is the best way to access Pharmadex 2 data from ETL (Extract, Transform, and Load) process?
   a) The usage of API calls to Pharmadex 2
   b) The usage of SQL queries, views, and stored procedures. It is the most appropriate approach.
   c) The development of special APIs

MODULE 2

1) The HL7 MedicinalProduct tree should be considered as a draft. Is it possible to use it in Pharmadex 2?
   a) No, it is impossible, because Pharmadex 2 should strictly obey the current HL7
   b) Yes, it is possible, because it is recommended to use the HL7 approach in case the HL7 standard is not available yet. The EMA thinks the same way.
2) Administrative feature "Resources" shows URLs. However, it is impossible to find these URLs in the view "tree_root". What do you think why?
   a) **These URLs are branches in some other trees. It is the right answer. Any concept is in a tree**.
   b) These URLs do not belong to any tree

MODULE 3

1) Why the field "Level" in the "closure" table is useful?
   a) This field is excess
   b) This field allows avoiding recursive SQL. This answer is the most appropriate.
   c) This field allows getting a root of a tree from any branch or leaf in a simple SQL query. This answer is also right

2) Is it possible to use only the “tree_root” view to explore a tree?
   a) Yes
   b) No. This view shows only the roots of the trees.
3) Is it possible to explore a tree from a leaf?
   a) Yes. The print_tree procedure allows it.
   b) No

Module 4

1) Suppose, that in some country is in use Gregorian and traditional calendar. Is it possible to keep dates separately?
   a) Yes. Theoretically, it is possible, however, the current input control does not allow any traditional calendar.
   b) No
2) Why does Pharmadex 2 use a concept with identifier _LITERALS_?
   a) It is the useless concept
   b) To distinct common use variables from others. Right answer. The _LITERALS_ creates a separate branch in a tree for the common use variables only.
   c) To improve the performance of SQL queries

Module 5

1) Who is responsible for application data configuration?
   a) Supervisor. Right answer. It is strongly unrecommended to re-programming the Pharmadex 2.
   b) Programmer
   c) Both
2) Which table are not in use for data configuration?
   a) concept
   b) closure
   c) activity
   d) assembly
3) Does the “clazz” column is available for a supervisor?
   a) Yes. It can be found in the user interface for the Supervisor
   b) No

Module 6

1) The ThingDoc object relates to concept dictNode. It is a concept of a dictionary item. Why do we need this reference?
   a) To provide a user a guide on which file should be uploadeRight answer. It is a common way how the file uploader works.
   b) To check file name against a dictionary item
   c) To attach the file to the dictionary
2) The query listed in Figure 26 returns more than one record. Is it possible?
   a) Yes. It is possible if the file uploader will ask to upload more than one file or there are many file uploaders.
   b) No
3) The query from Figure 26 returns zero records. Is it possible?
   a) Yes. The file uploader is not defined in the configuration.

b) No

MODULE 7

1) Is it possible that the application data fits in on-screen form?
   a) Yes. It should be a simple application
   b) No
2) The application data consists of the application data and sixteen additional data forms.
   a) Is it possible technically?
      i) Yes. Right answer. There is no limit for the additional data
      ii) No
   b) Is it acceptable?
      i) Yes
      ii) No. Right answer. Seventeen forms to fill out are completely unacceptable for a business user.
3) Who is solely responsible for application data configuration?
   a) Supervisor.
   b) Moderator
   c) Programmer
   d) Business User
4) Can I use the same "pharmacy.site.employees" additional data configuration for retail and wholesale pharmacies?
   a) Yes. It is a right answer
   b) No

MODULE 8

1) Is it possible to place a list of product manufacturers?
   a) Yes
   b) No. There is no pre-defined data component for this.
2) Is it possible to place a text field to the on-screen form that contains the "persons" pre-defined data component?
   a) Yes. There is any restriction to place any other pre-defined data component along with the "persons"
   b) No
3) Is it possible to use the pre-defined component "ingredients" in the additional data for medical devices?
   a) Yes. There is no restriction where to place "ingredients" and "persons"
   b) No
4) A pre-defined component consists of User Interface and server-side API. Does the "persons" component API use the ThingPerson ORM class in the server-side codes?
   a) Yes. It should make operation with this class.
   b) No
5) Can SQL query on Figure 38 return data for all employees
   a) Yes. This query is for this.
   b) No

MODULE 9

1) Can be INN in workflow data?

    a) Yes
    b) No. It is hard to imagine that an NMRA user will input INN instead of a business user.

2) Line 4 in SQL in Figure 42 contains criteria “clo.Level=2”. What will display in column actURL in case “clo.Level=1”?
    a) Error
    b) Owner’s eMail. Please, see it in Figure 41
    c) The ID of a concept
3) Line 6 contains the criteria “his.actConfigID is not null”. What does it mean?
    a) Some history records may not point to an activity of the workflow.
    b) Some activities have no configurations. Yes, it is possible. The null value is allowable for this field, see Figure 40

MODULE 10

1) Is it mandatory to follow SKOS “prefLabel” recommendations to configure any application data?
    a) Yes. It is presumed in the reports stored procedures. Additionally, it will be a good idea to get a uniform data source for the title of the application.
    b) No
2) Is it possible to get a list of all sorts of product applications using the report_products procedure?
    a) Yes
    b) No. The “report_products” procedure allows only one application data URL parameter.
3) Is it possible to create a uniform data query for all sorts of medical sites using the “report_sites” procedure?
    a) Yes
    b) No. The “report_sites” procedure allows only one application data URL parameter.
4) May I change the existing stored procedures?
    a) Yes
    b) **No. These procedures are used internally by Pharmadex 2. Create a copy with another name instead**.