OpenRIMS. Data visualization using Google Looker

(release 20240104)

Contents

Motivation

The OpenRIMS database contains two sets of tables and views. One set is adapted to data input and management, second is a Data Warehouse for data visualization

The content of the Data Warehouse is a result of the daily ETL[1] process against tables adapted for data input.

This manual contains examples of the typical data visualization use cases. The examples were created using Google Looker Studio. All examples are built on fictive data.

Annexes contain references to the Data Warehouses tables and views, an explanation of the permit lifecycle, and provide a recommended roadmap of the data visualization

To use this manual, the Google Looker knowledge is required. Please, check, do you understand such terms:

- Data Source
- Filter
- Dimension
- Metric
- Blend

This document deprecates all previous reporting and data visualization manuals.

[1] Extract, Transform, Load

Permit Register

Report

https://lookerstudio.google.com/u/0/reporting/745cf19d-2e72-4575-95d7-96e1cdb49557/page/VY3PD

The OpenRIMS is a permit issuance management electronic tool. Thus, a register of issued permits should be published. This example demonstrates the usage of

- The Data Warehouse view “report_permits”
- Filters
- OpenRIMS publicly available permit data feature [**Error! Reference source not found.**]

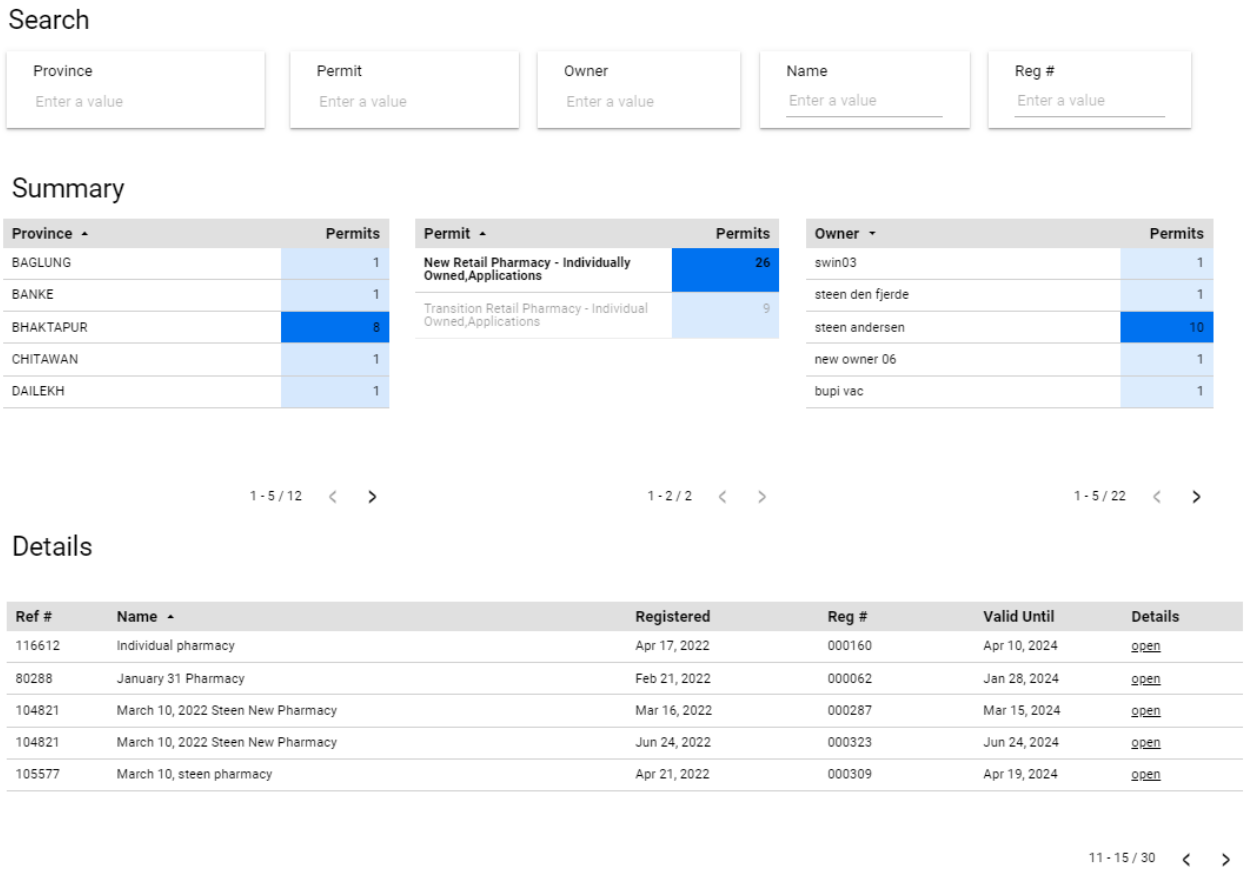

Figure 1 Permit Register

Filters

All these filters are for the whole report:

1. Language, to get all records in a particular language, e.g., EN_US:
   1.1. Lang=’EN_US’
2. Classifiers, to get the district from an address of the premise:
   2.1. ClassifierUrl= dictionary.admin.units and ClassifierLevel=1
3. Links to get applicants:

   3.1. LinkedURL contains own
4. Events to get incoming application registration numbers and dates:
   4.1. EventURL contains cert

Dimensions

Dimension	Field from report_permits
Province	ClassifierValue
Permit	PermitModule
Owner	Linked
Ref #	PermitID
Name	PermitName
Registered	EventDate
Reg #	EventRegNo
Valid Until	EventNextDate
Details	HYPERLINK(concat("https://pharmadex.irka.in.ua/public?lang=",Lang,"#publicpermitdata/%7B%22permitDataID%22:",PermitID,"%7D"), 'open')

Metrics

Metric	Formula
Permits	COUNT_DISTINCT(PermitID)

NMRA Workload KPI

Report

https://lookerstudio.google.com/u/0/reporting/f7809d67-d03c-40e8-a8c3-80fb2cef40df/page/29uND

The workload KPI report provides workload data for NRA departments and performance data for employees.

This example demonstrates the usage of

- The Data Warehouse view “report_kpi”
- Min, max, average metrics

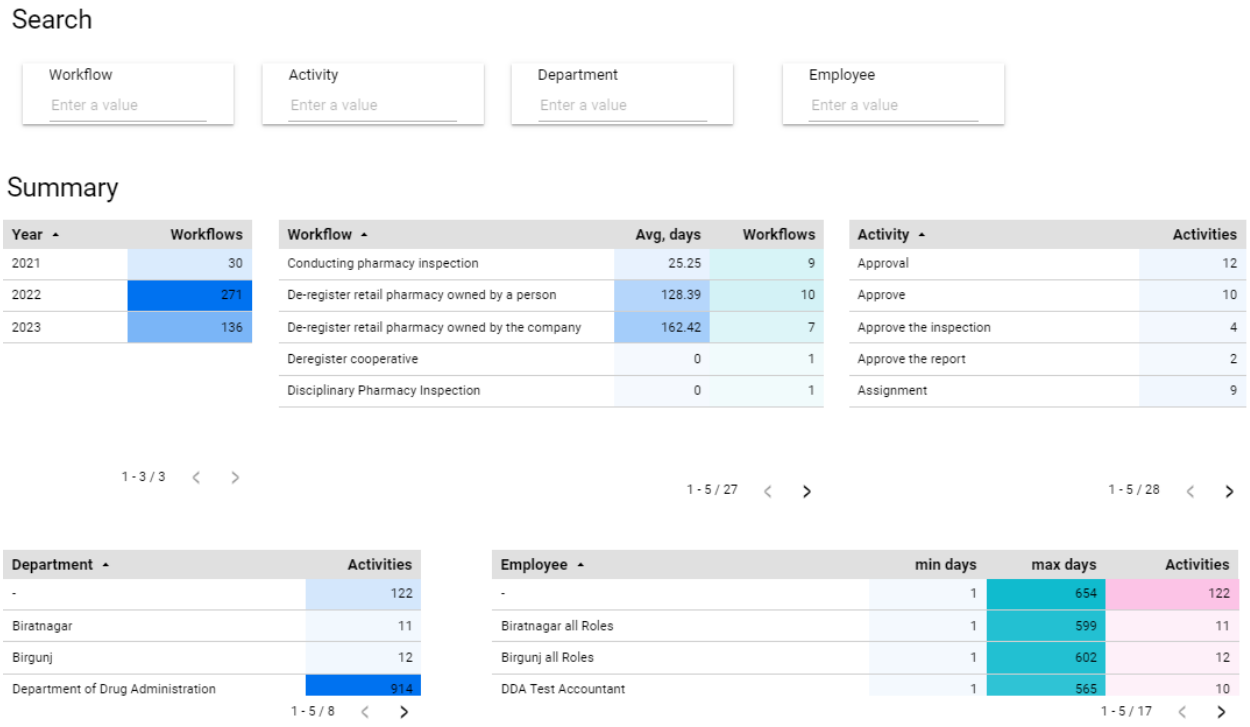

Figure 2 KPIs

Filters

1. Language, to get all records in a particular language, e.g., EN_US:
   1.1. Lang=’EN_US’

Dimensions

Dimension	Data from report_permits
Year	YEAR(WorkflowCome)
Workflow	Workflow
Activity	Activity
Department	ActivityDepartment
Employee	ActivityExecutorFullName

Metrics

Metric	Formula
Workflows	COUNT_DISTINCT(WorkflowID)
Avg Days	AVG(DATE_DIFF(WorkflowGo, WorkflowCome))
Activities	COUNT_DISTINCT(ActivityID)
Min Days	min(date_diff(date(ifnull(ActivityGo,TODAY())),ActivityCome))+1
Max Days	max(date_diff(date(ifnull(ActivityGo,TODAY())),ActivityCome))+1

Outcomes KPI

Report

https://lookerstudio.google.com/u/0/reporting/10412b6e-bbe7-4de8-92e2-3aa597ed1059/page/q9zVD

Any incoming application should be processed by NRA. The result of the processing is the one of outcomes:

- Approve
- Approve Company
- Decline

The analysis of outcomes is vital to improve the quality of NRA processes.

This example demonstrates the usage of the Data Warehouse view “report_kpi” as well as illustrates the state-transition diagram [The content of tables is a result of the daily ETL process against the transactional data. The views are built upon the tables to cover the typical use cases.

The ETL process is implemented by the stored procedure dwh_update that runs daily. To ensure stability and continuity, the Data Warehouse tables contain data collected by two ETL runs or sessions.

The views contain data for the latest session. The starting set of views is in the database. In case additional views are needed, it is possible to place them into a separate database to avoid the original database structure changes.

Tables

This manual does not cover the fields of tables. To understand the fields for any given table, examine them using the various SQL queries to a table.

#	Table	Purpose	Usage
1.	dwhactivities	To report activities in the workflows. An activity is a job executed by one person that ends up with an outcome.	This table will be useful to calculate KPIs – performance and workflow outcomes.
2.	dwhclassifiers	An electronic form may contain classifiers - dictionaries and drop lists. Examples are administrative units, types of business, etc.	To report classifier values. To calculate metrics based on the classifier values, e.g., quantity of premises in the province
3.	dwhdepartments	NRA employees by the departments	To report full names of employees and names of departments. To calculate metrics like quantity of activities done by an employee or a department
4.	dwhevents	A page in an electronic form may contain event definitions - registers and schedulers components.	To report event dates and numbers. To calculate metrics like “how many events…”

#	Table	Purpose	Usage
5.	dwhlinks	Sometimes it is necessary to define a 1:M relation from a permit to other data records, i.e., owners of a pharmacy, manufacturers of medicinal products, etc. The “persons” and “links” components are allowing this	To report names of linked data. To calculate metrics like “average quantity of owners” or “how many medicinal products manufactured by …”
6.	dwhpages	Contains all data pages related to this permit. Each page is defined as an electronic form	In SQL queries to join all classifiers, events, and links with any given permit
7.	dwhworkflowlist	All workflow configurations	To report names of workflows. To calculate metrics like “How many workflows of each type are defined”
8.	dwhworkflows	All workflows are completed and running	To report workflow dates, names, and permit names. To calculate various metrics
9.	reportclassifier	The deprecated version of dwhclassifiers	See dwhclassifiers
10.	reportdatamodules	List of the main pages of all applications. This table is deprecated by dwhpages	See dwhpages
11.	reportevent	The deprecated version of dwhevents	See dwhevents
12.	reportfullpreflabel	Following the W3C recommendations, the main human-readable name of any object or subject should be defined in a named data structure. The name of it is “prefLabel”.	In SQL queries to join all prefLabel to an electronic form page. Then, prefLabel will be used for the report An example can be found in the definition of view “dwhapplications” and others
13.	reportliteral	A typical application form is multi-page. A page may contain text fields (literals). A literal has a name. This table contains all literals on all application’s form pages	Get non-prefLabel literals from the application electronic form
14.	reportpage	The deprecated version of dwhworkflows. See also pages	See dwhworkflows
15.	reportpagelinks	The deprecated version of dwhlinks	See dwhlinks

#	Table	Purpose	Usage
16.	reportsession	To get access to the latest data warehouse	Only in SQL queries to ensure the right join to the other Data Warehouse tables. For example, to get the names of all applications: select distinct r.FullPrefLabel from history h **join reportsession rs on rs.Actual** join reportfullpreflabel r on r.conceptID=h.applDataID and **r.reportsessionid=rs.ID**
17.	pagesall	The deprecated version of dwhworkflows. See also reportpage	See dwhworkflows

Views

#	View	Purpose	Usage
1.	dwhapplications	Essential data for all applications and workflows	To create applications-related reports. An example is the [Incoming Applications Logbook]
2.	report_activities	The deprecated version of report_kpi	See report_kpi
3.	report_kpi	To report activities and workflow dates, durations, NRA departments, executors	To calculate KPIs. An example is [NMRA Workload KPI]
4.	report_permits	To report permits	Various permit registers. An example is [Permit Register]
5.	report_users	To limit access to the application, permits, and workflows using Google Login of the current user	Reports with limited access to data. An example is [Incoming Applications Logbook]

visualization.google.looker

Annex 2 States and transitions].

Please note, that the workflow outcome and permit state are not the same.

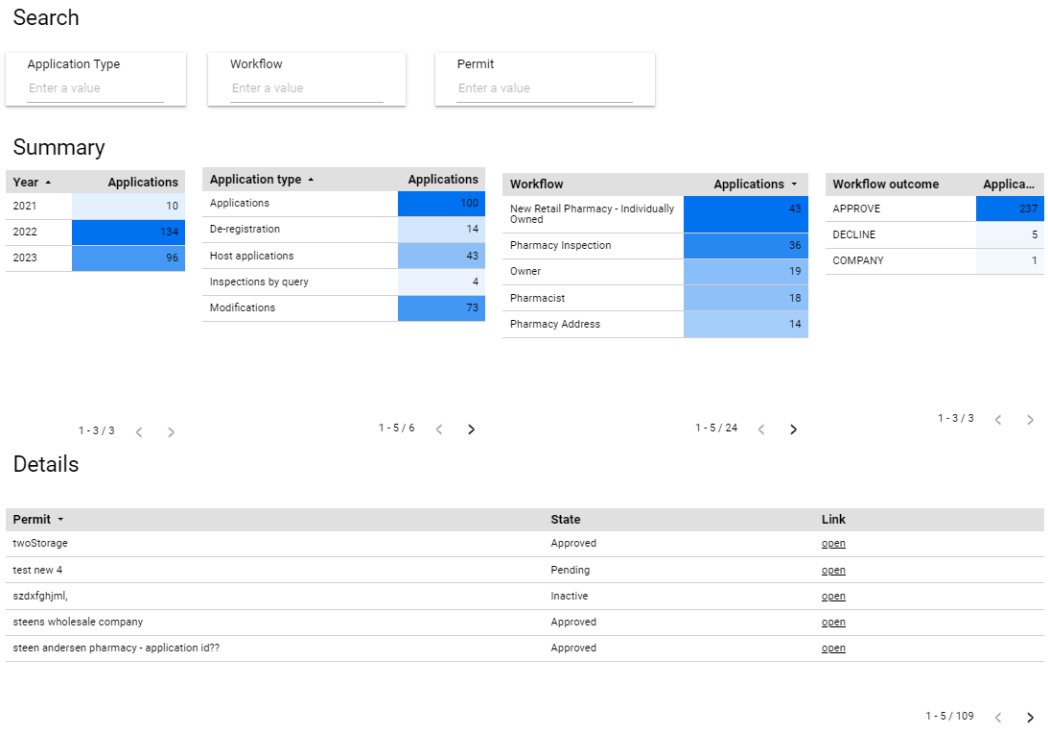

Figure 3 Outcome KPI

Filters

1. Language, to get all records in a particular language, e.g., EN_US:
   1.1. Lang='EN_US'
2. ActivityOutcome to exclude non-terminal activities:
   2.1. ActivityOutcome in {COMPANY, APPROVE, DECLINE}

Metrics

Metric	Formula
Applications	COUNT_DISTINCT(WorkflowID)

Dimensions

Dimension	Data from report_permits
Year	YEAR(WorkflowGo)
Application Type	WorkflowDictName
Workflow	Workflow
Workflow outcome	ActivityOutcome
Permit	PermitName
State	case PermitState when 'ACTIVE' then 'Approved' when 'ONAPPROVAL' then 'Pending' when 'NOTSUBMITTED' then 'Not submitted'

	when 'DEREGISTERED' then 'Inactive' END
Link	HYPERLINK(concat("https://pharmadex.irka.in.ua/public?lang=",Lang,"#publicpermitdata/%7B%22permitDataID%22:",PermitID,"%7D"), 'open')

Incoming Applications Logbook

Report

https://lookerstudio.google.com/u/0/reporting/1883d3bc-81ad-42c0-ba2e-2e0cdbbf7387/page/L7ScD

The Incoming Application Logbook is an example of a digital incoming documents logbook

This example demonstrates the usage of:

- Data Warehouse view “dwhapplications” to get the application’s data
- Data Warehouse view “report_users”
- Access control using the Google login
- Google Looker Studio blends
- Registration numbers
- Registration, Deadline, or Expiration dates

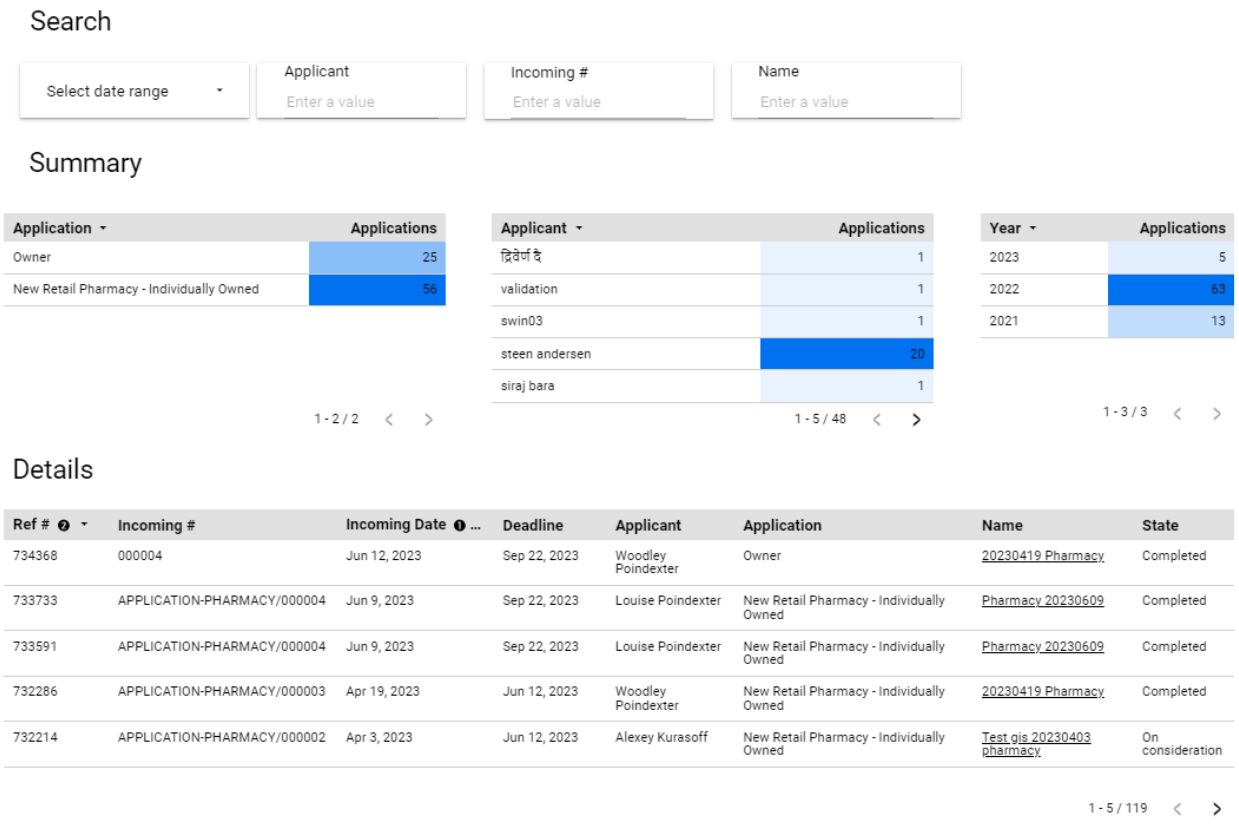

Figure 4 Incoming Application Logbook

Access Control

The Data Warehouse view “report_users” intends to use OpenRIMS application access rules in Google Looker reports.

Column	Usage
ApplDictID	Unique identifier of a kind of application, i.e., “Medicine Marketing Authorization”, “Pharmacy Business Allowance”
DictUrl	URL of the applications dictionary in OpenRIMS, i.e., dictionary.guest.applications, dictionary.guest.deregistration
Email	The Google email of the NRA employee for which this kind of application is accessible. This dimension is declared as “Filter by email” in the data source

In Google Looker, this view should be defined as a data source that filters data by email

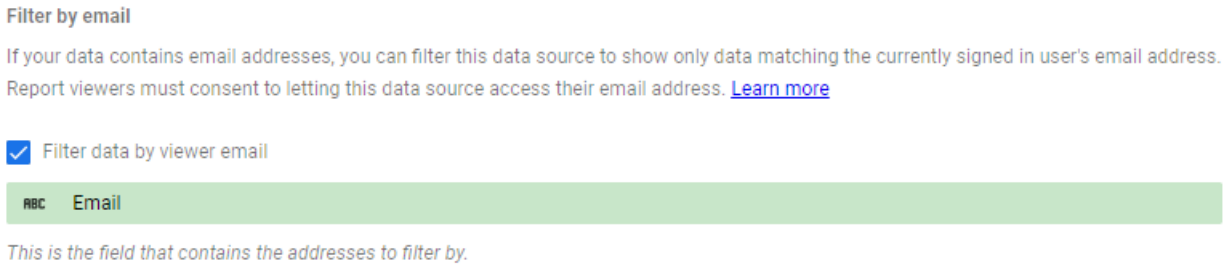

Figure 5 In the data source declare the Email field as "Filter data by viewer email"

Blend

The data source is the database view “dwhapplications”. To allow access control the data source should be joined with the data source report_users on dimension ApplDictID. In terms of Google Looker Studio, this construct is called “blend” To create the blend:

1. Add both data sources to the report
2. Create a blend with join as below
3. Extract to the blend all necessary dimensions from “dwhapplications”

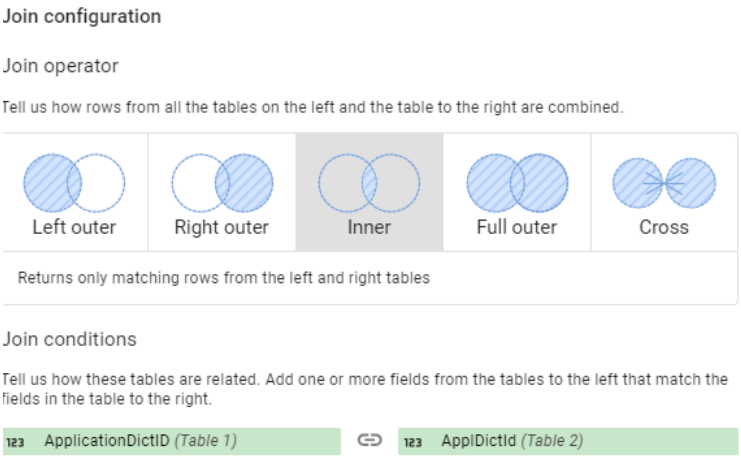

Figure 6 The join should be inner

Filters

All filters are defined in the blend.

1. Language, to get all records in a particular language, e.g., EN_US:
   1.1. Lang='EN_US'
2. EventURL, to get incoming application registration records from the application's events:
   2.1. EventURL in { register.new.pharmacy, register.amendments.pharmacy, register.pharmacy.deregistration}
3. LinkedPageURL, to get applicants:
   3.1. LinkedPageURL= site.owner.person

Dimensions

Dimension	Data from dwhapplications
Application	ApplicationDictPref
Applicant	Linked
Year	Year(EventDate)
Ref #	ApplicationID
Incoming	EventNumber
Incoming Date	EventDate
Deadline	IFNULL(NextEventDate, TODAY())
Applicant	Linked
Name	HYPERLINK(concat("https://pharmadex.irka.in.ua/public?lang=",'EN_US',"#publicpermitdata/%7B%22permitDataID%22:",PermitDataID,"%7D"), ApplicationPrefLabel)
State	if(Completed=1,'Completed','On consideration')

Metrics

Metric	Formula
Applications	COUNT_DISTINCT(ApplicationID)

Annex 1 The Data Warehouse

Overview

The Data Warehouse consists of database tables and views. The content of tables is a result of the daily ETL process against the transactional data. The views are built upon the tables to cover the typical use cases.

The ETL process is implemented by the stored procedure dwh_update that runs daily. To ensure stability and continuity, the Data Warehouse tables contain data collected by two ETL runs or sessions.

The views contain data for the latest session. The starting set of views is in the database. In case additional views are needed, it is possible to place them into a separate database to avoid the original database structure changes.

Tables

This manual does not cover the fields of tables. To understand the fields for any given table, examine them using the various SQL queries to a table.

#	Table	Purpose	Usage
18.	dwhactivities	To report activities in the workflows. An activity is a job executed by one person that ends up with an outcome.	This table will be useful to calculate KPIs – performance and workflow outcomes.
19.	dwhclassifiers	An electronic form may contain classifiers - dictionaries and drop lists. Examples are administrative units, types of business, etc.	To report classifier values. To calculate metrics based on the classifier values, e.g., quantity of premises in the province
20.	dwhdepartments	NRA employees by the departments	To report full names of employees and names of departments. To calculate metrics like quantity of activities done by an employee or a department
21.	dwhevents	A page in an electronic form may contain event definitions - registers and schedulers components.	To report event dates and numbers. To calculate metrics like “how many events…”
22.	dwhlinks	Sometimes it is necessary to define a 1:M relation from a permit to other data records, i.e., owners of a pharmacy, manufacturers of medicinal products, etc. The “persons” and “links” components are allowing this	To report names of linked data. To calculate metrics like “average quantity of owners” or “how many medicinal products manufactured by …”

#	Table	Purpose	Usage
23.	dwhpages	Contains all data pages related to this permit. Each page is defined as an electronic form	In SQL queries to join all classifiers, events, and links with any given permit
24.	dwhworkflowlist	All workflow configurations	To report names of workflows. To calculate metrics like “How many workflows of each type are defined”
25.	dwhworkflows	All workflows are completed and running	To report workflow dates, names, and permit names. To calculate various metrics
26.	reportclassifier	The deprecated version of dwhclassifiers	See dwhclassifiers
27.	reportdatamodules	List of the main pages of all applications. This table is deprecated by dwhpages	See dwhpages
28.	reportevent	The deprecated version of dwhevents	See dwhevents
29.	reportfullpreflabel	Following the W3C recommendations, the main human-readable name of any object or subject should be defined in a named data structure. The name of it is “prefLabel”.	In SQL queries to join all prefLabel to an electronic form page. Then, prefLabel will be used for the report An example can be found in the definition of view “dwhapplications” and others
30.	reportliteral	A typical application form is multi-page. A page may contain text fields (literals). A literal has a name. This table contains all literals on all application’s form pages	Get non-prefLabel literals from the application electronic form
31.	reportpage	The deprecated version of dwhworkflows. See also pages	See dwhworkflows
32.	reportpagelinks	The deprecated version of dwhlinks	See dwhlinks
33.	reportsession	To get access to the latest data warehouse	Only in SQL queries to ensure the right join to the other Data Warehouse tables. For example, to get the names of all applications: select distinct r.FullPrefLabel from history h **join reportsession rs on rs.Actual** join reportfullpreflabel r on r.conceptID=h.applDataID and **r.reportsessionid=rs.ID**

#	Table	Purpose	Usage
34.	pagesall	The deprecated version of dwhworkflows. See also reportpage	See dwhworkflows

Views

#	View	Purpose	Usage
6.	dwhapplications	Essential data for all applications and workflows	To create applications-related reports. An example is the [Incoming Applications Logbook][2]
7.	report_activities	The deprecated version of report_kpi	See report_kpi
8.	report_kpi	To report activities and workflow dates, durations, NRA departments, executors	To calculate KPIs. An example is [NMRA Workload KPI][3]
9.	report_permits	To report permits	Various permit registers. An example is [Permit Register][4]
10.	report_users	To limit access to the application, permits, and workflows using Google Login of the current user	Reports with limited access to data. An example is [Incoming Applications Logbook][5]

[2] https://lookerstudio.google.com/u/0/reporting/1883d3bc-81ad-42c0-ba2e-2e0cdbbf7387/page/L7ScD

[3] https://lookerstudio.google.com/u/0/reporting/f7809d67-d03c-40e8-a8c3-80fb2cef40df/page/29uND

[4] https://lookerstudio.google.com/u/0/reporting/745cf19d-2e72-4575-95d7-96e1cdb49557/page/VY3PD

[5] https://lookerstudio.google.com/u/0/reporting/1883d3bc-81ad-42c0-ba2e-2e0cdbbf7387/page/L7ScD

Annex 2 States and transitions

To make proper data visualization it will be necessary to understand the lifecycle of a permit.

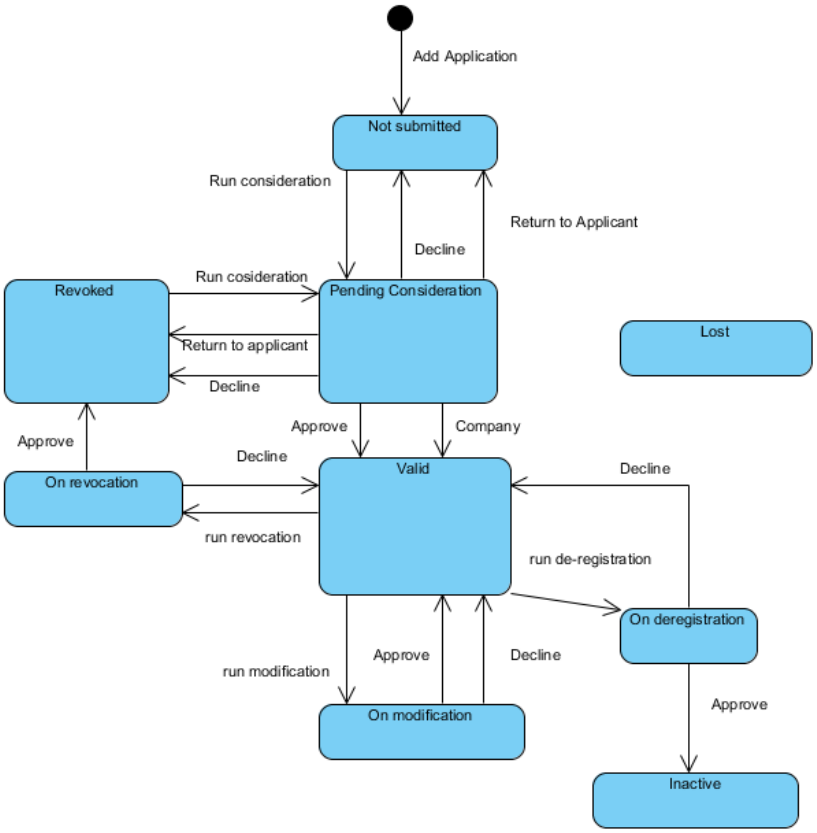

Figure 7 Permit life cycle. States and transitions

- The main path of states is “Not Submitted” -> “Pending Considerations” -> “Active” –> “Inactive”.
- States “On Modification”, “On revocation”, and “On deregistration” are sub-states of the “Valid” state. Currently, there is no way to distinguish them in reports.
- To follow the main path in the forward direction the respective workflows should be started
- Any given workflow has two possible outcomes – Approve and Decline. In case the outcome is “Approved” the permit will go into the next state. In case of the outcome “Decline,” the permit will go into the previous state.
- The “Pending Consideration” state allows two additional outcomes:
    - Return To Applicant – return the application to the applicant to fix minor mistakes
    - Company – to approve the company registration application
- The “Lost” state is reserved for permits that do not fit any other state, because of software errors.

Annex 3 Data Visualization Roadmap

Create the wireframe

The wireframe should be created to roughly determine dimensions and metrics, thus data sources.

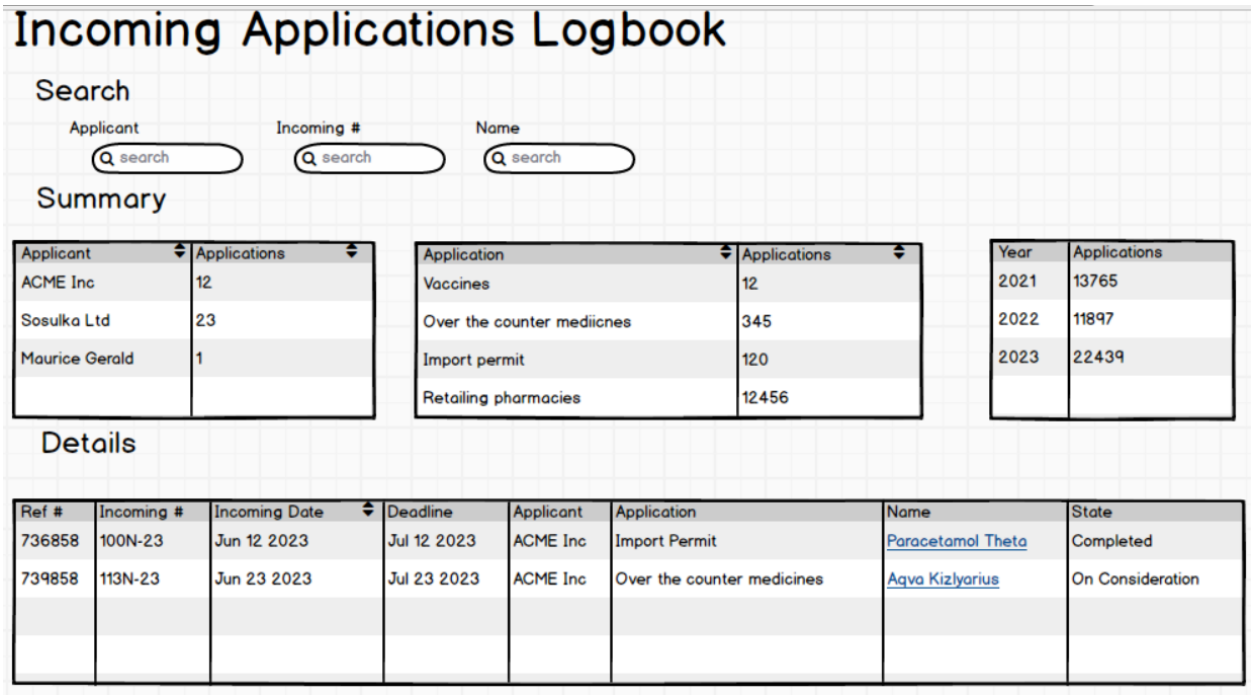

Figure 8 An example of the wireframe

Select or create a Report Data Source

It is presumed, that the dimensions and metrics have been determined by the wireframe

First, inspect the Data Warehouse views for necessary dimensions.

If there aren't appropriate views, create an SQL query for the data source. Make the query as common as possible. To limit data, report filters are preferable to the phrase "were" in the SQL query.

Decide access limitation

If the report is not publicity available, decide on two access control options:

- Limit access to the whole report using the Share feature of the Google Looker Studio
- Limit access to reporting data using the report_users view

In the case of the usage of report_users, it will be necessary to create a data blend similar to [Blend]

Create data filters

It will be a good idea to place data filters to report and/or data blend levels. Use the filters on a chart only if necessary.

Create data visualization

To create a good data visualization, please refer to appropriate books. We like a book by Claus Wilke "Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures"[6]

Place additional controls and decors

The most used additional controls are search boxes. The implementation in the Google Looker Studio is the Input box Control. This control allows exact and partial search by any dimension.

Other controls also can be very helpful. Please, refer to the Google Documentation.

The report may be decorated with a header and footer that allows imitation of OpenRIMS UI.

[6] https://www.amazon.com/Fundamentals-Data-Visualization-Informative-Compelling/dp/1492031089