DUY TÂN UNIVERSITY
International School

CAPSTONE PROJECT 2

CMU-SE-451 / CMU-IS-451 / CMU-CS-451

DATABASE DESIGN

Version 2.0

Date: 1 - Mar - 2021

EXPERT-DRIVEN SMART DASHBOARD APPLICATION

Submitted by

Vo Van Hoa
Pham Van Tin
Ky Huu Dong
Tran Thi Thanh Kieu

Approved by

Capstone Project 2 - Mentor:

Name	Signature	Date
Binh, Thanh Nguyen		26 - May - 2020

PROJECT INFORMATION			
Project Acronym	EDSDA		
Project Title	Expert-Driven Smart Dashboard Application		
Project Web URL	https://sda-research.ml/		
Start Date	01 - Mar - 2021		
End Date:	02 - Jun - 2021		
Lead Institution	International School, Duy Tan University		
Project Mentor	Ph.D Binh, Thanh Nguyen		
Scrum Master	Hoa, Vo	hoavo.dng@gmail.com	0935.193.182
Team Members	Tin, Pham Van	tinphamvan123@gmail.com	0932.535.175
	Dong, Ky Huu	kyhuudong@gmail.com	0898.246.980
	Kieu, Tran Thi Thanh	thanhkieutran391@gmail.com	0358.583.251

DOCUMENT INFORMATION			
Document Title	Database Design		
Author(s)	Team C2SE.06		
Role	[EDSDA] DatabaseDesign_v2.0		
Date	01 - Mar - 2021	Filename	[EDSDA] 006 Database Design
URL	https://github.com/sdateamdtu2020/SDA-v2.0		
Access	Project and CMU Program		

REVISION HISTORY

Version	Person(s)	Date	Description	Approval
Draft	Hoa, Vo	12 - Aug - 2020	Initiate document	x
2.0	All members	14 - Nov - 2020	Finish content of document	x

TABLE OF CONTENTS

1. INTRODUCTION

The Database Design maps the logical data model to the target database management system with consideration to the system's performance requirements. The Database Design converts logical or conceptual data constructs to physical data constructs (e.g tables,...) of the target Database Management System.

1.1. PURPOSE

The purpose of the Database Design is to ensure that every database transaction meets or exceeds its performance requirements. This document takes into account data and transaction volume to produce a schema and environment that will meet necessary performance

1.2. DOCUMENT OBJECTIVES

The Database Design Document has the following objectives:

- To describe the design of a database, that is, a collection of related data stored in one or more computerized files that can be accessed by users or developers via a DBMS
- To serve as a basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information necessary for software development.

1.3. INTENDED AUDIENCE

This document is intended for the following audiences:

Technical developers, who must evaluate the quality of this document

Developer including:

- Architects, whose overall architecture design must meet the requirements specified in this document.
- Designers, whose design must meet the requirements specified in this document.
- Developers, whose software must implement the requirements specified in this document.
- Quality Assurance personnel, whose test cases must validate the requirements specified in this document.

1.4. SCOPE, APPROACH AND METHODS

The Database Design for the SDA is composed of definitions for database objects derived by mapping entities to tables, attributes to columns, unique identifiers to unique keys and relationships to foreign keys.

1.5. SYSTEM OVERVIEW

System Overview	Details
System Name	SMART DASHBOARD APPLICATION
System type	Web application
Operational status	In development

1.6. ACRONYMS AND ABBREVIATIONS

Acronym/Abbreviation	Meaning
SDA	**S**MART **D**ASHBOARD **A**PPLICATION
PG	**P**ost**g**res
RDMS	**R**elational **D**atabase **M**anagement **S**ystem
DBMS	**D**ata**b**ase **M**anagement **S**ystem
RDC	**R**DF **D**ata **C**ube

1.7. KEY PERSONNEL

Full Name	Email	Phone number	Role
Hoa, Vo	hoavo.dng@gmail.com	0935.193.182	Scrum master
Tin, Pham Van	tinphamvan123@gmail.com	0932.535.175	Team member
Dong, Ky Huu	kyhuudong@gmail.com	0898.246.980	Team member
Kieu, Tran Thanh Thi	thanhkieutran391@gmail.com	0358.583.251	Team member

2. ASSUMPTIONS, CONSTRAINTS AND DEPENDENCIES

2.1. ASSUMPTIONS

- Users can drag the widgets to generate charts, maps, or any available solution on GUI.
- Users can link nodes to the others nodes to generate charts based on their needs.

2.2. CONSTRAINTS

- Users can only use all the functions available on GUI.

3. SYSTEM OVERVIEW

3.1. DATABASE MANAGEMENT SYSTEM CONFIGURATION

- **System**: GraphDB
- **Vendor**: Ontotext
- **Services enabled**:
    - RDF Data storage.
- **System**: Linux VM
- **Vendor**: Google
- **Services enabled**:
    - Linux Virtual Machine

3.2. DATABASE SOFTWARE UTILITIES

Vendor	Product	Version	Comments
pgAdmin	PGAdmin	4.0.0	This application enables the ability to mage the cloud PG from local machine

3.3. SUPPORT SOFTWARE

Product	Version	Purpose
DBeaver	Offline software	Help showing the table itself and its relations with other tables inside the schema on PostgreSQL platform

4. ARCHITECTURE

4.1. HARDWARE & SOFTWARE ARCHITECTURE

Handled by GraphDB

4.2. DATASTORES

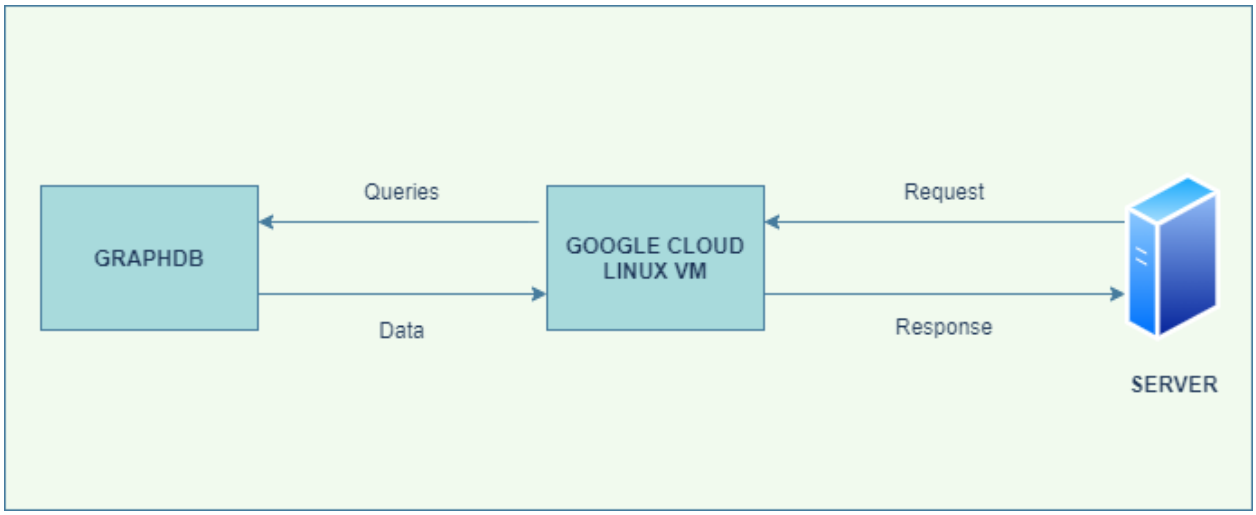


Figure 1. Datastores

GraphDB is a family of well-functioning, robust and awesome RDF databases. It reduces the load and use of connected cloud databases, as well as the RDF resources. We use GraphDB on a Linux Virtual Machine that is runned on Google Cloud platform for better querying, high performance processing and faster response.

5. DATABASE-WIDE DESIGN DECISIONS

5.1. KEY FACTORS INFLUENCING DESIGN

- The database should be designed independent when the frontend and backend are still being developed.
- The database should be designed to meet the data warehouse principles.

5.2. PERFORMANCE AND AVAILABILITY DECISIONS

- We use common dimensional tables for all the fact tables for better performance.
- The data warehouse should be designed with the star schema for speed-up querying and processing time.
- The data warehouse should be redundant for reducing response time and meet up with the backend design pattern of the system.
- Data warehouse should be stored with PostgreSQL.
- RDF Data Cubes should be fully defined.
- RDF Data Cubes should be deployed on GraphDB.
- Class hierarchy and relationships should be done before extracting the data.
- RDF Data Cubes should be validated correctly before writing RestAPI.

6. DATABASE ADMINISTRATIVE FUNCTIONS

6.1. RESPONSIBILITY

Role	Name	Responsibility	Email Address
Database Administrator	Dong, Ky Huu		kyhuudong@gmail.com
System Administrator	Hoa, Vo		hoavo.dng@gmail.com
Security Administrator	Hoa, Vo		hoavo.dng@gmail.com

6.2. APPLICATIONS/SYSTEMS USING THE DATABASE

System ID	Model	Version	System Code
EDSDA-Web	NA	In Development	NA

6.3. PHYSICAL DESIGN

6.3.1. FACT-INDUSTRY

factindustry				
Field	**Type**	**Constraint**	**Nullable**	**Description**
Industryid	int(auto increment)	PK	No	Industry id is primary key to specific unique row
areaid	VARCHAR	FK	No	area id is the foreign key and is the primary key of the area dimension
yearid	int	FK	No	Year id is the foreign key and is the primary key of the city dimension
industry	double		Yes	Industrial data
description	text		Yes	Description for each unique row id

6.3.2. FACT-POPULATION

factpopulation				
Field	**Type**	**Constraint**	**Nullable**	**Description**
populationid	int(auto increment)	PK	No	Population id is primary key to specific unique row
areaid	VARCHAR	FK	No	area id is the foreign key and is the primary key of the area dimension
yearid	int	FK	No	Year id is the foreign key and is the primary key of the city dimension
population	double		Yes	Population data
description	text		Yes	Description for each unique row id

6.3.3. FACT-CLIMATE

factclimate				
Field	**Type**	**Constraint**	**Nullable**	**Description**
climateid	int(auto increment)	PK	No	Climate id is primary key to specific unique row
areaid	VARCHAR	FK	No	area id is the foreign key and is the primary key of the area dimension
yearid	int	FK	No	Year id is the foreign key and is the primary key of the city dimension
humidity	double		Yes	Humidity data
rainfall	double		Yes	Rainfall data
temperature	double		Yes	Temperature data
description	text		Yes	Description for each unique row id

6.3.4. FACT-FOREST

factforest				
Field	**Type**	**Constraint**	**Nullable**	**Description**
forestid	int(auto increment)	PK	No	Forest id is primary key to specific unique row
areaid	VARCHAR	FK	No	area id is the foreign key and is the primary key of the area dimension
yearid	int	FK	No	Year id is the foreign key and is the primary key of the city dimension
afforestation	double		Yes	Afforestation data
sumofforestacreage	double		Yes	Forest acreage data
ratioforestcover	double		Yes	The cover ratio of forest data
description	text		Yes	Description for each unique row id

6.3.5. DIMAREA

dimcity				
Field	**Type**	**Constraint**	**Nullable**	**Description**
areaid	VARCHAR	PK	No	Area id is primary key to specific unique row
area	VARCHAR		No	Area name data
cityid	VARCHAR		No	City id is the foreign key and is the primary key of the city dimension

6.3.6. DIMCITY

dimcity				
Field	**Type**	**Constraint**	**Nullable**	**Description**
cityid	VARCHAR	PK	No	City id is primary key to specific unique row
city	VARCHAR		No	City name data
districtid	VARCHAR		NO	District id is the foreign key and is the primary key of the district dimension

6.3.7. DIMDISTRICT

dimcity				
Field	**Type**	**Constraint**	**Nullable**	**Description**
districtid	VARCHAR	PK	No	District id is primary key to specific unique row
district	VARCHAR		No	District name data

6.3.8. DIMYEAR

dimyear				
Field	**Type**	**Constraint**	**Nullable**	**Description**
yearid	int(auto increment)	PK	No	Year id is primary key to specific unique row
year	int		No	Year data

6.3.9. FOREST-SOURCE-DATA-STAGING

forestsourcedatastaging				
Field	**Type**	**Constraint**	**Nullable**	**Description**
Source_stagingid	int(auto increment)	PK	No	Source_staging id is primary key to specific unique row
area	VARCHAR		Yes	Area name data
areaid	VARCHAR		Yes	Area id from dimarea
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
district	VARCHAR		Yes	District name data
districtid	VARCHAR		Yes	District id from dimdistrict
year	int		Yes	Year data
yearid	int		Yes	Year id from dimyear
afforestation	double		Yes	Afforestation data
sumofforestacreage	double		Yes	The forest acreage data
ratioforestcover	double		Yes	The cover ratio of forest data

6.3.10. CLIMATE-SOURCE-DATA-STAGING

climatesourcedatastaging				
Field	**Type**	**Constraint**	**Nullable**	**Description**
source_stagingid	int(auto increment)	PK	No	Source_staging id is primary key to specific unique row
area	VARCHAR		Yes	Area name data
areaid	VARCHAR		Yes	Area id from dimarea
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
district	VARCHAR		Yes	District name data
districtid	VARCHAR		Yes	District id from dimdistrict
year	int		Yes	Year data
yearid	int		Yes	Year id from dim year
humidity	double		Yes	Humidity data
rainfall	double		Yes	Rainfall data
temperature	double		Yes	Temperature data

6.3.11. POPULATION-SOURCE-DATA-STAGING

populationsourcedatastaging				
Field	**Type**	**Constraint**	**Nullable**	**Description**
source_stagingid	int(auto increment)	PK	No	Source_staging id is primary key to specific unique row
area	VARCHAR		Yes	Area name data
areaid	VARCHAR		Yes	Area id from dimarea
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
district	VARCHAR		Yes	District name data
districtid	VARCHAR		Yes	District id from dimdistrict
year	int		Yes	Year data
yearid	int		Yes	Year id from dimyear
population	double		Yes	Population data

6.3.12. INDUSTRY-SOURCE-DATA-STAGING

industrysourcedatastaging				
Field	**Type**	**Constraint**	**Nullable**	**Description**
source_stagingid	int(auto increment)	PK	No	Source_staging id is primary key to specific unique row
area	VARCHAR		Yes	Area name data
areaid	VARCHAR		Yes	Area id from dimarea

Field	Type	Constraint	Nullable	Description
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
district	VARCHAR		Yes	District name data
districtid	VARCHAR		Yes	District id from dimdistrict
year	int		Yes	Year data
yearid	int		Yes	Year id from dimyear
industry	double		Yes	Industrial data

6.4. RDF Data Cubes Design

6.4.1. Dimension

Column	Description
city	The area belongs to dataset
cityid	ID name transformed from city data
year	The time period belongs to dataset

6.4.2. Measure

Column	Description
humidity	Observed humidity value
rainfall	Observed rainfall value
temperature	Observed temperature value
forestarea	Observed forest area value
forestcover	Observed forest cover value

Column	Description
deforestation	Observed deforestation value
naturalforestarea	Observed natural forest area value
industry	Observed industrial value
population	Observed population value

6.4.3. Structure, Patterns, and Local Prefixes

6.4.3.1. Climate Data Cube

Table 1. Structure, Patterns, Prefixes

Item **[prefix]** {pattern} Description	**Value for Project**
Cube Name (Dataset name)	climate
BaseURI	http://sda-research.ml/
Data Cube {BaseURI}dc/{cube name}	http://sda-research.ml/dc/climate
DataSet **[ds]** {BaseURI}dc/{cube name}/dataset Includes the qb:DataSet, the qb:DataStructureDefinition and the qb:Observation. The values of each dimension (specified as the value of the cube property in each dimension as part of qb:Observation) are also placed here because they are values that are a part of the cube. This would change if codelists are used. Slices [qb:Slice, qb:SliceKey] would also be included here, if used.	http://sda-research.ml/dc/climate/dataset

Properties **[prop]** {BaseURI}dc/{cube name}/prop/ Properties of the Data Cube. a) **qb:ComponentProperty, qb:DimensionProperty, qb:MeasureProperty, qb:AttributeProperty,** qb:CodedProperty b) **qb:component** defined under each Data	http://sda-research.ml/dc/climate/prop/
Cube Component Specifications **[dccs]** {BaseURI}dc/{cube name}/**dccs**/ Cube Component specifications. **qb:ComponentSpecification**	http://sda-research.ml/dc/climate/dccs/

Table 2. URI's for Dimensions and Measures

Component Pattern	Value for Project
dimension	1. http://sda-research.ml/dc/climate/prop/city 2. http://sda-research.ml/dc/climate/prop/cityid 3. http://sda-research.ml/dc/climate/prop/year
measure	1.http://sda-research.ml/dc/climate/prop/humidity 2. http://sda-research.ml/dc/climate/prop/rainfall 3.http://sda-research.ml/dc/climate/prop/temperatur-e

6.4.3.2. Industry Data Cube

Table 1. Structure, Patterns, Prefixes

Item [prefix] {pattern} Description	Value for Project
Cube Name (Dataset name)	Industry
BaseURI	http://sda-research.ml/
Data Cube	http://sda-research.ml/dc/industry

{BaseURI}dc/{cube name}	
DataSet **[ds]** {BaseURI}dc/{cube name}/dataset Includes the qb:DataSet, the qb:DataStructureDefinition and the qb:Observation. The values of each dimension (specified as the value of the cube property in each dimension as part of qb:Observation) are also placed here because they are values that are a part of the cube. This would change if codelists are used. Slices [qb:Slice, qb:SliceKey] would also be included here, if used.	http://sda-research.ml/dc/industry/dataset
Properties **[prop]** {BaseURI}dc/{cube name}/prop/ Properties of the Data Cube. a) **qb:ComponentProperty, qb:DimensionProperty, qb:MeasureProperty, qb:AttributeProperty,** qb:CodedProperty b) **qb:component** defined under each Data	http://sda-research.ml/dc/industry/prop/
Cube Component Specifications **[dccs]** {BaseURI}dc/{cube name}/**dccs**/ Cube Component specifications. **qb:ComponentSpecification**	http://sda-research.ml/dc/industry/dccs/

Table 2. URI's for Dimensions and Measures

Component Pattern	Value for Project
dimension	1. http://sda-research.ml/dc/industry/prop/city 2. http://sda-research.ml/dc/industry/prop/cityid 3. http://sda-research.ml/dc/industry/prop/year
measure	1.http://sda-research.ml/dc/industry/prop/humidity

6.4.4. External vocabularies

Prefix	URI	Comment
qb	http://purl.org/linked-data/cube#	Cube spec.
rdfs	http://www.w3.org/2000/01/rdf-schema#	Labels, comments
xsd	http://www.w3.org/2001/XMLSchema#	Data types
dcat	http://www.w3.org/ns/dcat#	Distribution information
dct	http://purl.org/dc/terms/	Creator, issued date, title, description...
prov	http://www.w3.org/ns/prov#	Provenance

6.4.5. Class hierarchy

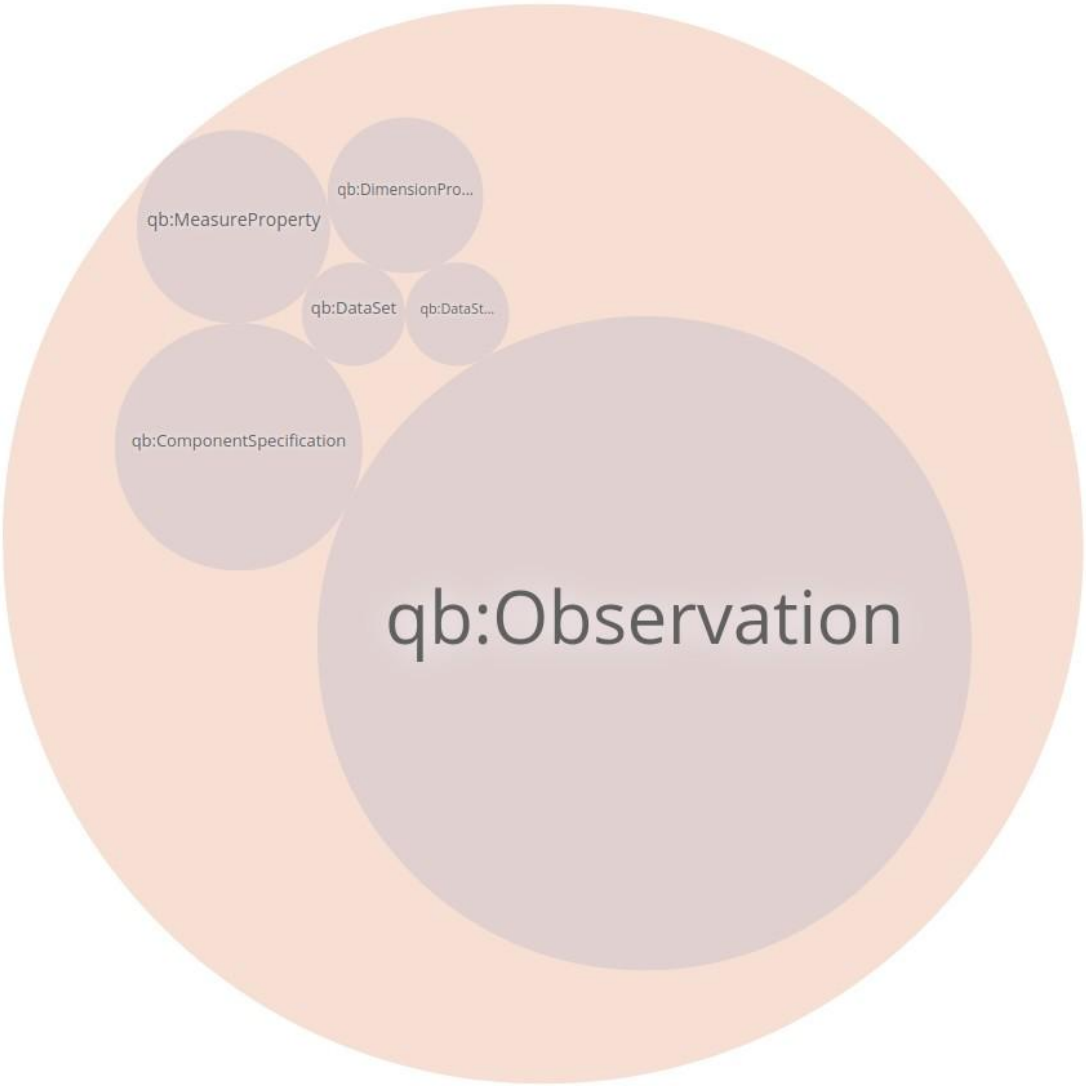


Figure 2. Class Hierarchy

6.4.6. Class relationships

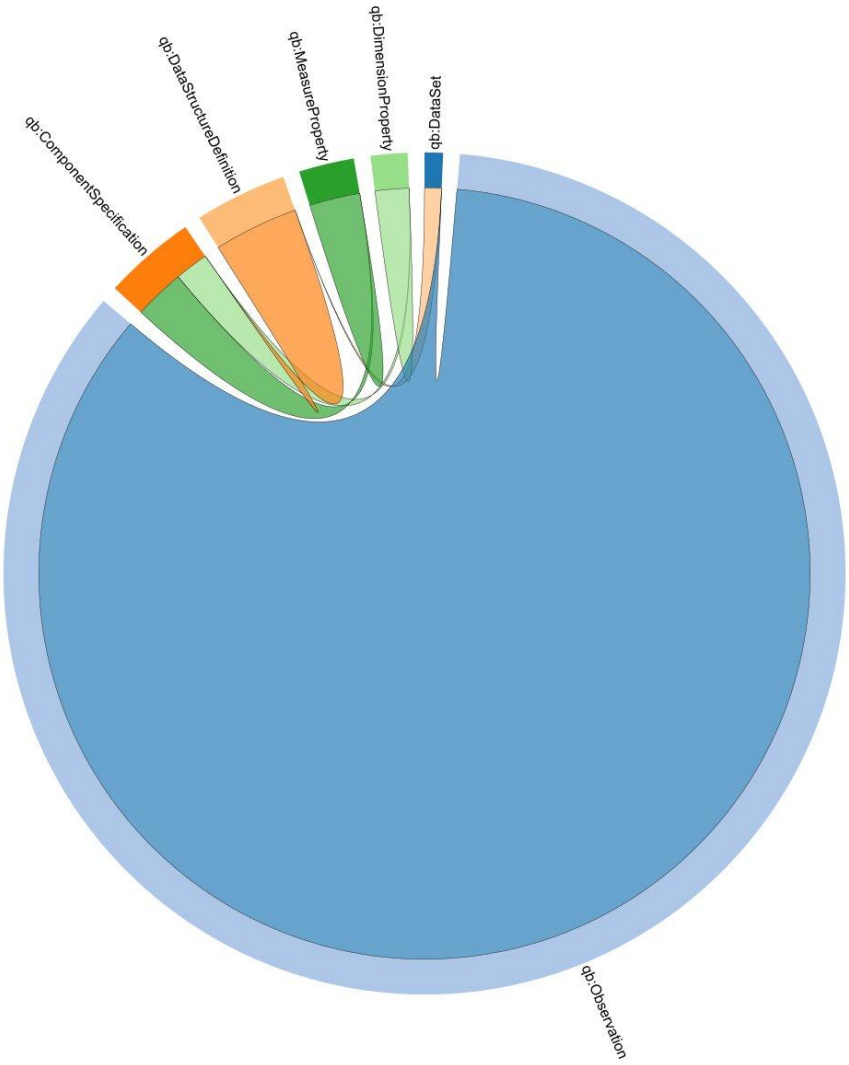


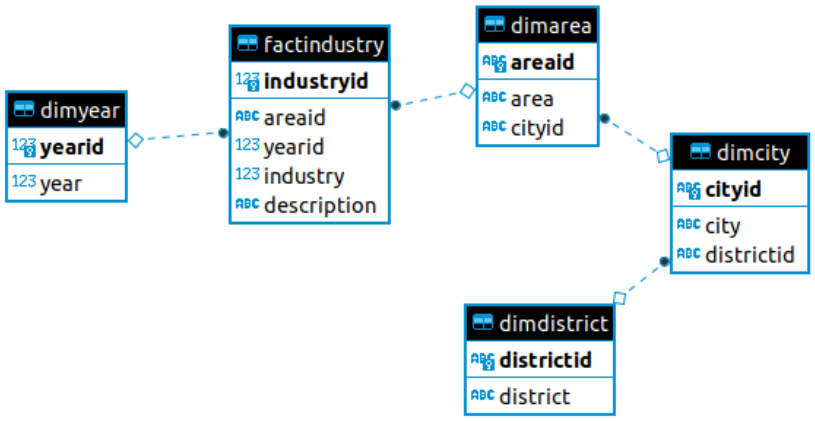
Figure 3. Class Relationships

6.5. ENTITY MAPPING

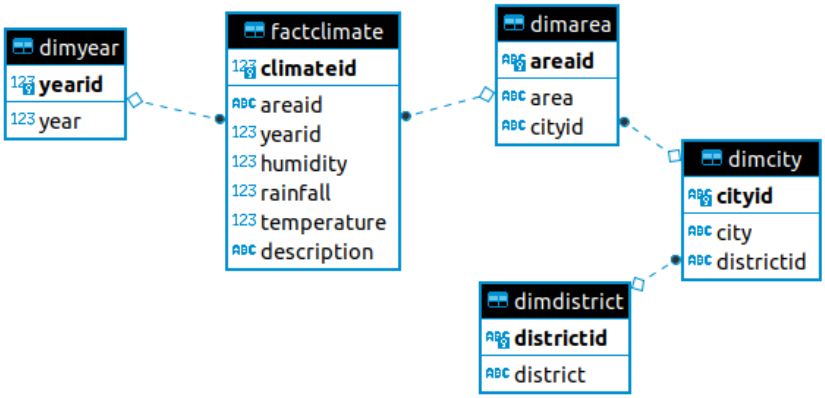
6.5.1. ENTITY MAPPING DIAGRAM

Please see the attached image for more details.

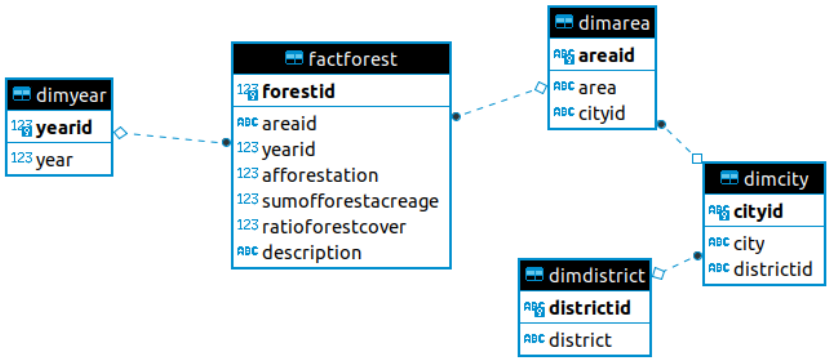
6.5.2. INDUSTRY



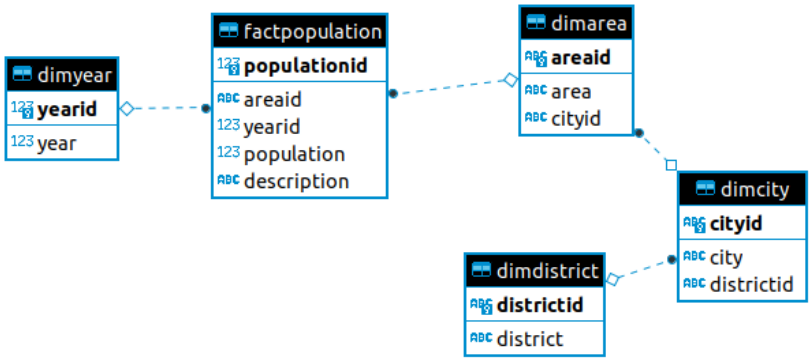
6.5.3. CLIMATE



6.5.4. FOREST



6.5.5. POPULATION



7. REFERENCES

- Technologies Stack Document
- dbdiagram.io : https://dbdiagram.io
- The RDF Data Cube: https://www.w3.org/TR/eo-qb/#Datacube
- Entity Relationship Mapping: https://docs.oracle.com/cd/A97688_16/generic.903/a97677/ormap.htm
- Entity Mapping Diagram For modeling ETL processes: https://www.researchgate.net/profile/Ali_El-Bastawissy/publication/236030320_Entity_Mapping_Diagram/links/00463515e17cb192b4000000/Entity-Mapping-Diagram.pdf