



CAPSTONE PROJECT 1

CMU-SE-450 / CMU-IS-450 / CMU-CS-450

DATABASE DESIGN DOCUMENT

Version 1.2

Date: 12 - Aug - 2020

SMART DASHBOARD APPLICATION

Submitted by

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

Approved by

Capstone Project 1 - Mentor:

Name

Signature

Date

Binh, Thanh Nguyen _____

A handwritten signature in blue ink, appearing to read 'Nguyen Binh Thanh', written over a horizontal line.

_____ 14 - Dec- 2020

Name

Signature

Date

Huy, Truong Dinh _____

PROJECT INFORMATION			
Project Acronym	SDA		
Project Title	Smart Dashboard Application		
Project Web URL	https://sda-research.ml/		
Start Date	12 - Aug - 2020		
End Date:	15 - Dec - 2020		
Lead Institution	International School, Duy Tan University		
Project Mentor	PhD Binh, Nguyen Thanh; MSc Huy, Truong Dinh		
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 Thanh	thanhkieutran391@gmail.com	0358.583.251

DOCUMENT INFORMATION			
Document Title	Database Design Document		
Author(s)	Team C1SE.06		
Role	[SDA] Database_Design_v1.2		
Date	21 - Nov - 2020	File name	[SDA] Database_Design_v1.2
URL	https://github.com/sdateamdtu2020/sda-documents		
Access	Project and CMU Program		

REVISION HISTORY

Version	Person(s)	Date	Description	Approval
Draft	Hoa, Vo	12 - Aug - 2020	Initiate document	x
1.0	All members	14 - Nov - 2020	Finish content of document	x
1.1	Hoa, Kieu, Dong	16 - Nov - 2020	Update content of Physical Design	x
1.2	Hoa, Dong	21 - Nov - 2020	Update RDF Data Cubes Design & Entity Mapping	

TABLE OF CONTENTS

PROJECT INFORMATION	1
DOCUMENT INFORMATION	1
REVISION HISTORY	2
TABLE OF CONTENTS	3
1. INTRODUCTION	5
1.1. PURPOSE	5
1.2. DOCUMENT OBJECTIVES	5
1.3. INTENDED AUDIENCE	5
1.4. SCOPE, APPROACH AND METHODS	6
1.5. SYSTEM OVERVIEW	6
1.6. ACRONYMS AND ABBREVIATIONS	6
1.7. KEY PERSONNEL	6
2. ASSUMPTIONS, CONSTRAINTS AND DEPENDENCIES	7
2.1. ASSUMPTIONS	7
2.2. CONSTRAINTS	7
3. SYSTEM OVERVIEW	7
3.1. DATABASE MANAGEMENT SYSTEM CONFIGURATION	7
3.2. DATABASE SOFTWARE UTILITIES	7
3.3. SUPPORT SOFTWARE	7
4. ARCHITECTURE	8
4.1. HARDWARE & SOFTWARE ARCHITECTURE	8
4.2. DATASTORES	8
5. DATABASE-WIDE DESIGN DECISIONS	9
5.1. KEY FACTORS INFLUENCING DESIGN	9
5.2. PERFORMANCE AND AVAILABILITY DECISIONS	9
6. DATABASE ADMINISTRATIVE FUNCTIONS	9
6.1. RESPONSIBILITY	9
6.2. APPLICATIONS/SYSTEMS USING THE DATABASE	10
6.3. PHYSICAL DESIGN	10
6.3.1. FACT-INDUSTRY	10

6.3.2. FACT-POPULATION	10
6.3.3. FACT-CLIMATE	11
6.3.4. FACT-FOREST	12
6.3.5. DIMCITY	12
6.3.6. DIMYEAR	13
6.3.7. FOREST-SOURCE-DATA-STAGING	13
6.3.8. CLIMATE-SOURCE-DATA-STAGING	13
6.3.9. POPULATION-SOURCE-DATA-STAGING	14
6.3.10. INDUSTRY-SOURCE-DATA-STAGING	14
6.4. RDF DATA CUBES DESIGN	15
6.4.1. DIMENSION	15
6.4.2. MEASURE	15
6.4.3. STRUCTURE, PATTERNS AND LOCAL PREFIXES	16
6.4.4. EXTERNAL VOCABULARIES	19
6.4.5. CLASS HIERARCHY	20
6.4.6. CLASS RELATIONSHIPS	21
6.5. ENTITY MAPPING	21
6.5.1. ENTITY MAPPING DIAGRAM	21
6.5.2. INDUSTRY	22
6.5.3. CLIMATE	22
6.5.4. FOREST	23
6.5.5. POPULATION	23
7. REFERENCES	24

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 PPPLICATION
PG	P ost g res
RDMS	R elational D atabase M anagement S ystem
DBMS	D atabase 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, map, 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 7.2.0	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

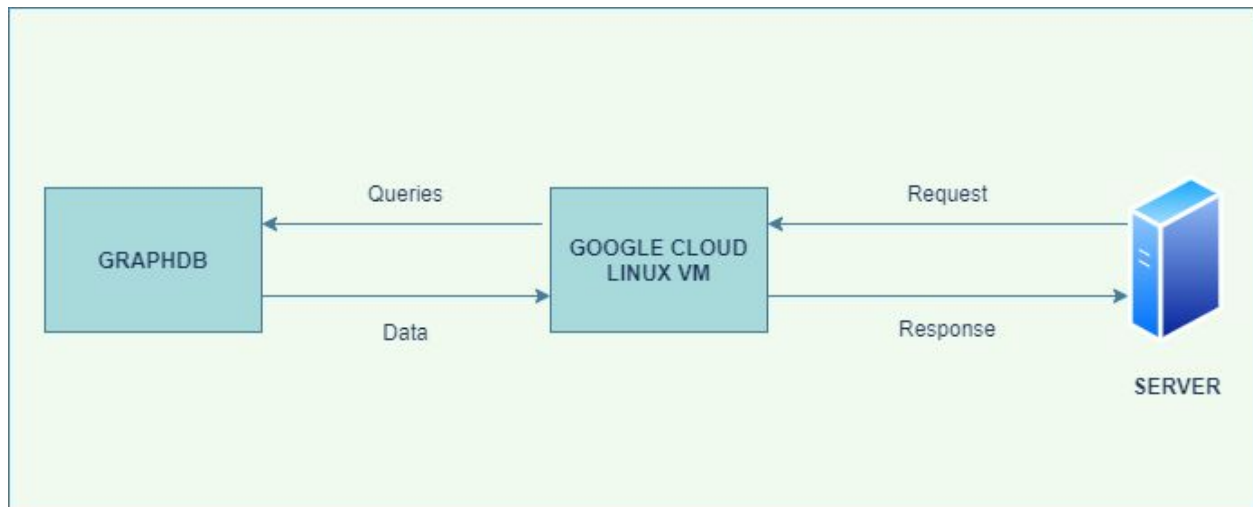


Figure 1. Datastores

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 an 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
SDA-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
cityid	VARCHAR	FK	No	City id is the foreign key and is the primary key of the city 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

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
cityid	VARCHAR	FK	No	City id is the foreign key and is the primary key of the

				city 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

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
cityid	VARCHAR	FK	No	City id is the foreign key and is the primary key of the city 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

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
cityid	VARCHAR	FK	No	City id is the foreign key and is the primary key of the city 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
forestcover	double		Yes	Forest Cover data

6.3.5. DIMCITY

dimcity				
Field	Type	Constraint	Nullable	Description
cityid	VARCHAR(SEQUENCE('xxxx'), 1000)	PK	No	City id is primary key to specific unique row
city	VARCHAR		No	City name data

6.3.6. 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

6.3.7. 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
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
year	int		Yes	Year data
yearid	int		Yes	Year id from dimyear
afforestation	double		Yes	Afforestation data
forestcover	double		Yes	Forestcover data

6.3.8. 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
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
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

6.3.9. 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
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
year	int		Yes	Year data
yearid	int		Yes	Year id from dimyear
population	double		Yes	Population data

6.3.10. 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
city	VARCHAR		Yes	City name data
cityid	VARCHAR		Yes	City id from dimcity
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
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	http://sda-research.ml/dc/climate/dataset

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.	
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 [dccc] {BaseURI}dc/{cube name}/dccc/ Cube Component specifications. qb:ComponentSpecification	http://sda-research.ml/dc/climate/dccc/

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/temperature

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 {BaseURI}dc/{cube name}	http://sda-research.ml/dc/industry
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 [dcs] {BaseURI}dc/{cube name}/dcs/ Cube Component specifications. qb:ComponentSpecification	http://sda-research.ml/dc/industry/dcs/

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

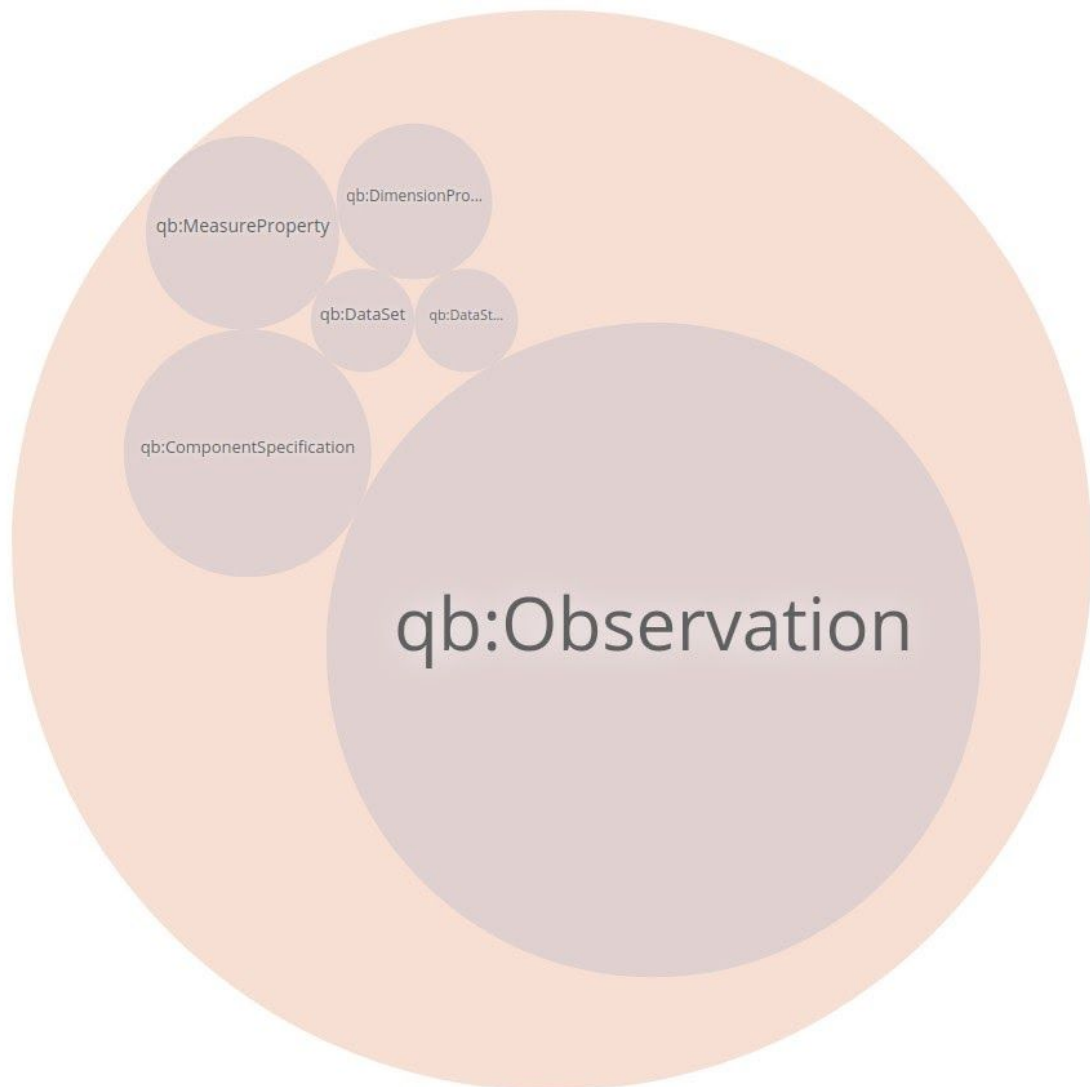


Figure 2. Class Hierarchy

6.4.6. Class relationships

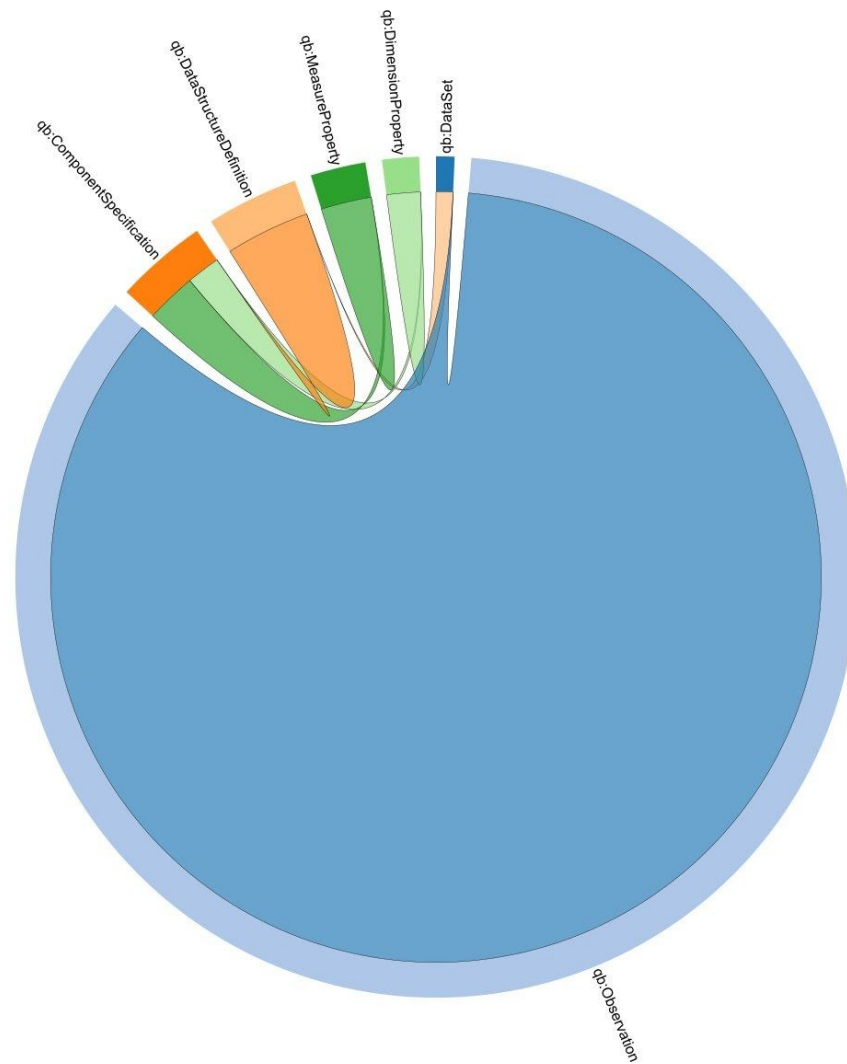


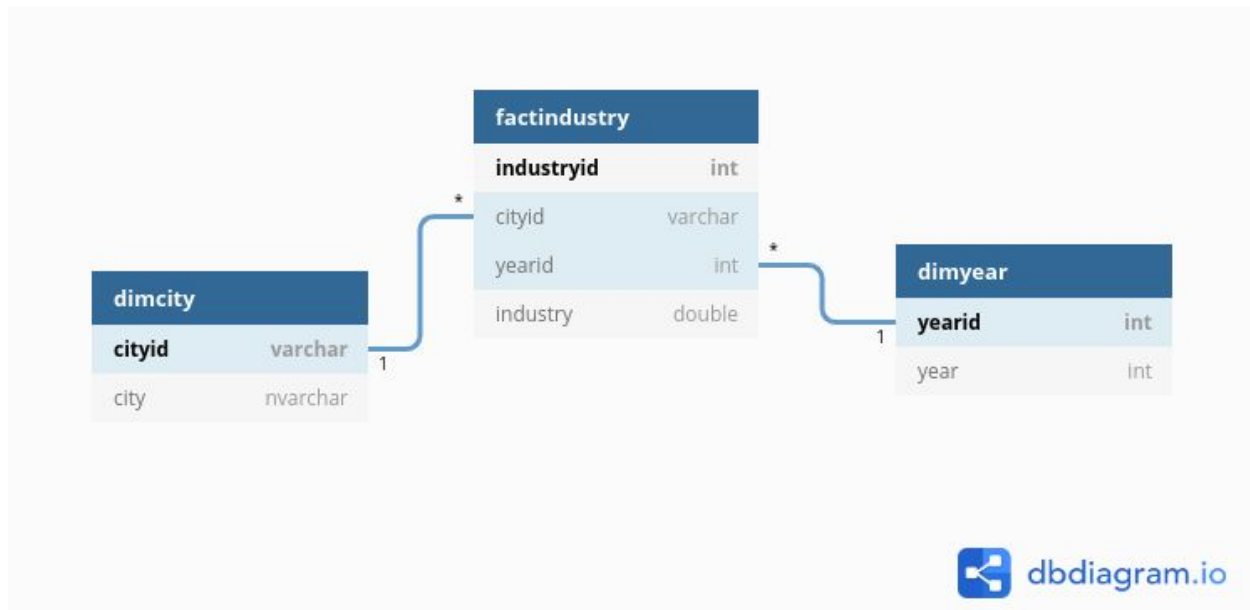
Figure 3. Class Relationships

6.5. ENTITY MAPPING

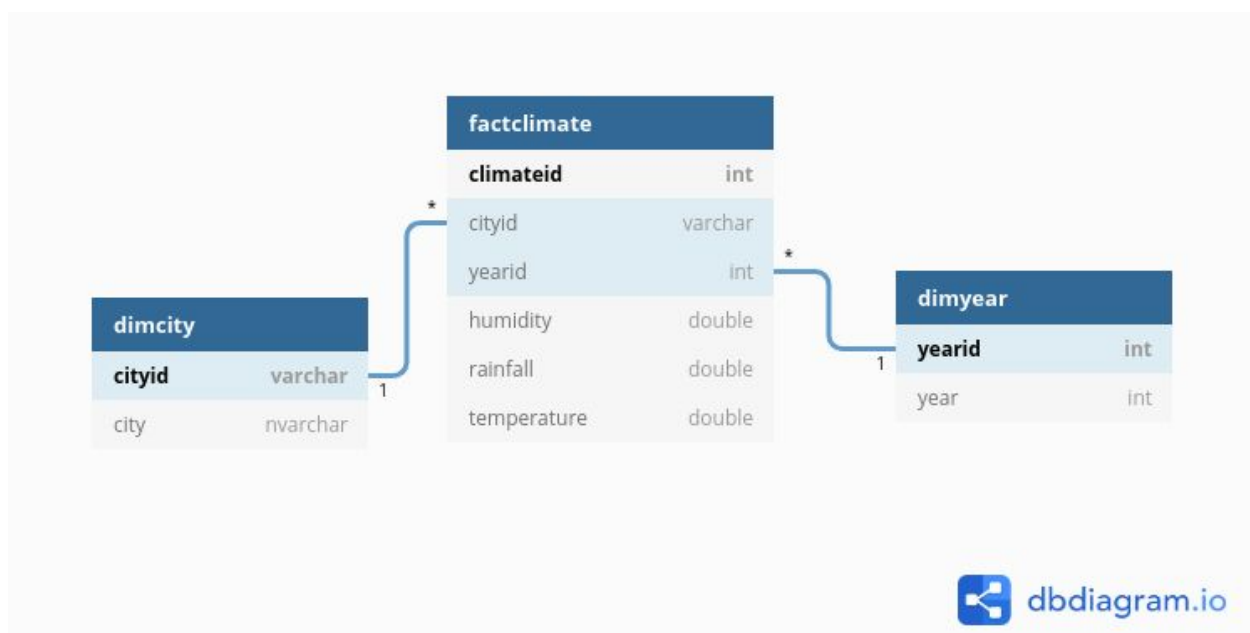
6.5.1. ENTITY MAPPING DIAGRAM

- Please see the attached image for more details.

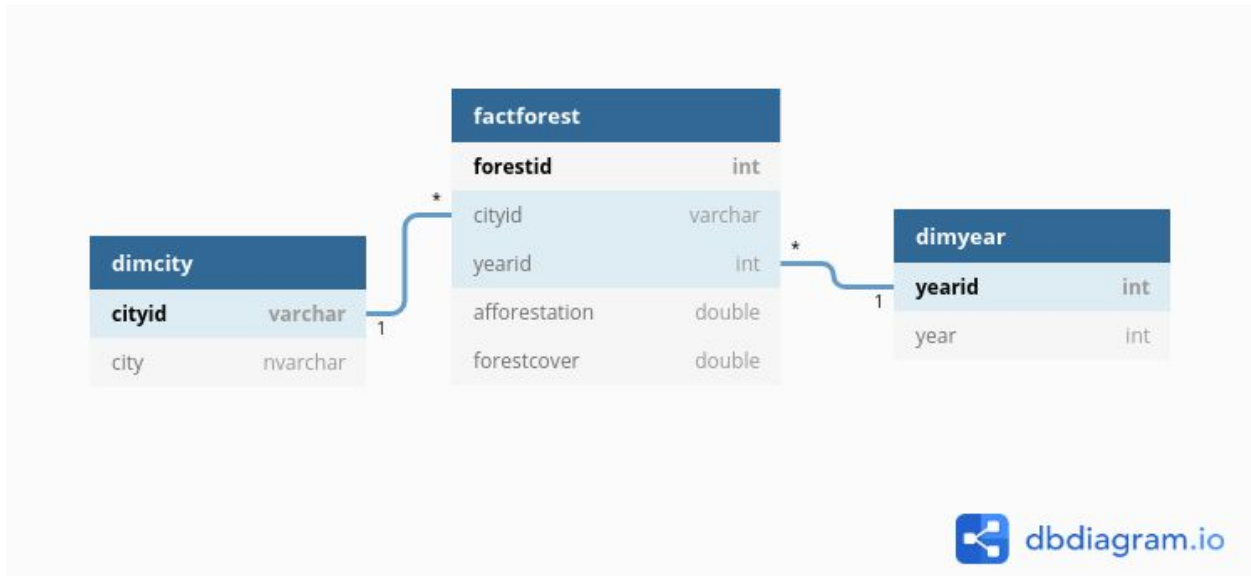
6.5.2. INDUSTRY



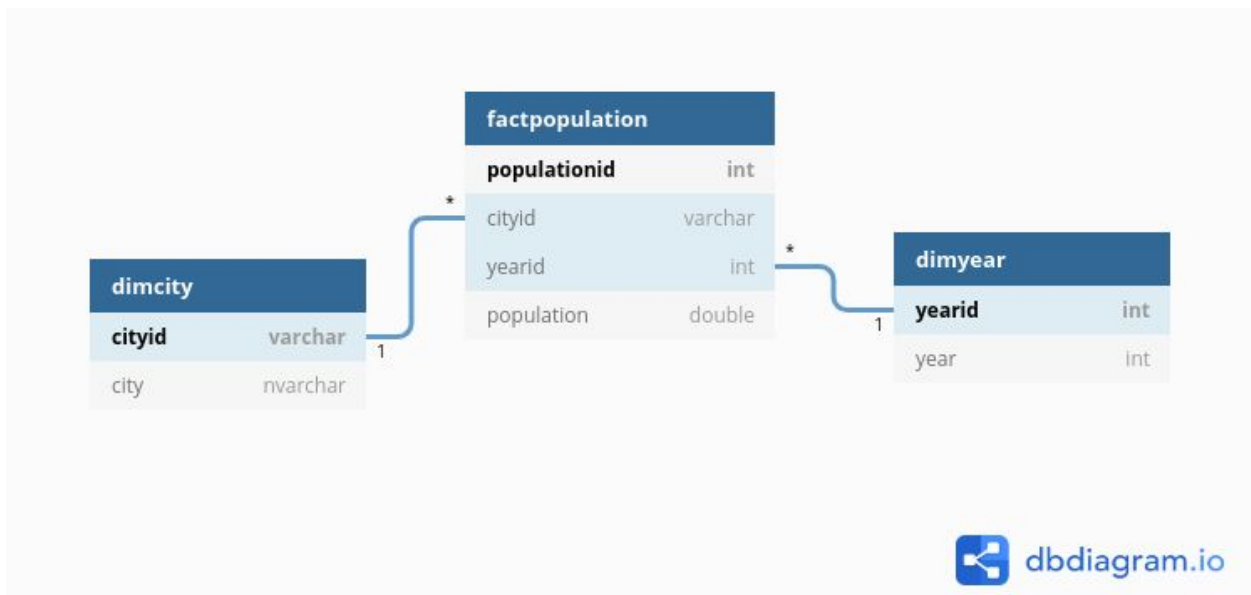
6.5.3. CLIMATE



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