###Text

Text z\_cheatsheet::links::cheatsheet\_link

{

type: plainText;

content: 'https://github.com/finos/legend/wiki/GHC-Cheat-Sheet';

}

###Diagram

Diagram model::EmissionsModelDiagram

{

classView 0a227d35-7600-4c1a-85a5-d3063abc6fde

{

class: model::GasProductionBasedCarbonDioxideEmissions;

position: (1050.5707797604866,339.94375882817627);

rectangle: (287.392578125,86.0);

}

classView 17ec1cd8-b2c0-406b-9783-083aef4fd7af

{

class: model::CementProductionBasedCarbonDioxideEmissions;

position: (1049.425658507277,453.09135126365885);

rectangle: (287.380859375,86.0);

}

classView 9292d69e-9823-43ba-8b29-015cd721753f

{

class: model::CoalProductionBasedCarbonDioxideEmissions;

position: (1051.5921279732086,566.3027255882855);

rectangle: (288.0,86.0);

}

classView ff8874d5-1f53-406a-8e85-0687297efe2b

{

class: model::FlaringBasedCarbonDioxideEmissions;

position: (1052.508893240243,685.5063417240776);

rectangle: (287.0,104.0);

}

classView 09e02f77-893b-449f-8fcb-27cc1bde37fd

{

class: model::LandUseBasedCarbonDioxideEmissions;

position: (1053.2849289629053,812.3146923527258);

rectangle: (284.0,91.0);

}

classView 0a519a0e-bf27-483e-82bb-52312df07cab

{

class: model::OilProductionBasedCarbonDioxideEmissions;

position: (1057.5231760981746,931.1705963283571);

rectangle: (280.712890625,86.0);

}

classView a50e24c0-6c01-4ea6-8bf0-823ed1f6a921

{

class: model::OtherIndustryCarbonDioxideEmissions;

position: (1061.039566157518,1032.8020708139502);

rectangle: (281.0,100.0);

}

classView 44d5da05-ca29-4187-87d9-69c0820409b5

{

class: model::Country;

position: (-347.012067050179,158.74962290468187);

rectangle: (397.7490234375,305.5);

}

classView a66434bc-303b-4f39-a04f-a748c47658c8

{

class: model::ProductionBasedCarbonDioxideEmissions;

position: (527.5281127887845,331.97952936060904);

rectangle: (291.2939453125,792.6253628026508);

}

classView 6e412d99-3996-4d4d-825a-14eddd935ca7

{

class: model::CarbonDioxideEmissions;

position: (547.0572388477069,168.5154925680954);

rectangle: (244.9521484375,100.0);

}

classView 98ebae4c-2406-4f70-a6ce-5f813d7ef400

{

class: model::SurfaceTemperature;

position: (171.99580670668144,160.73791709477416);

rectangle: (267.21622174006484,178.1441478267099);

}

propertyView

{

property: model::Country\_ProductionBasedCarbonDioxideEmissions.country;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 44d5da05-ca29-4187-87d9-69c0820409b5;

points: [(552.8833052011084,372.7683171543477),(46.49715735967362,375.23143065701026)];

}

propertyView

{

property: model::Country\_ProductionBasedCarbonDioxideEmissions.carbonDioxideEmissions;

source: 44d5da05-ca29-4187-87d9-69c0820409b5;

target: a66434bc-303b-4f39-a04f-a748c47658c8;

points: [(25.996403169037478,435.02529704636584),(550.6785058609064,434.5026986800061)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.gasProductionBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 0a227d35-7600-4c1a-85a5-d3063abc6fde;

points: [(792.1040336130351,380.48511484505497),(1194.2670688229866,382.94375882817627)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.cementProductionBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 17ec1cd8-b2c0-406b-9783-083aef4fd7af;

points: [(778.8752375718226,494.03228086546255),(1193.116088194777,496.09135126365885)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.flaringBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: ff8874d5-1f53-406a-8e85-0687297efe2b;

points: [(763.441642190408,734.3554089474901),(1196.008893240243,737.5063417240776)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.coalProductionBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 9292d69e-9823-43ba-8b29-015cd721753f;

points: [(772.2608395512162,607.5794468858701),(1195.5921279732086,609.3027255882855)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.landUseBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 09e02f77-893b-449f-8fcb-27cc1bde37fd;

points: [(766.748841200711,852.3121736483018),(1195.2849289629053,857.8146923527258)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.oilProductionBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 0a519a0e-bf27-483e-82bb-52312df07cab;

points: [(756.8272441698017,969.1665386790125),(1197.8796214106746,974.1705963283571)];

}

propertyView

{

property: model::ProductionBasedCarbonDioxideEmissions.otherIndustryBasedCO2Emissions;

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: a50e24c0-6c01-4ea6-8bf0-823ed1f6a921;

points: [(763.441642190408,1075.5481068437634),(1201.539566157518,1082.8020708139502)];

}

propertyView

{

property: model::SurfaceTemperature.country;

source: 98ebae4c-2406-4f70-a6ce-5f813d7ef400;

target: 44d5da05-ca29-4187-87d9-69c0820409b5;

points: [(176.85428346559178,231.99557622545814),(11.666073662642589,231.99557622545814)];

}

propertyView

{

property: model::Country.surfaceTemperature;

source: 44d5da05-ca29-4187-87d9-69c0820409b5;

target: 98ebae4c-2406-4f70-a6ce-5f813d7ef400;

points: [(8.427089156702436,251.42948326109922),(52.153379986894834,304.8727276091122),(197.90768275420294,303.25323535614206)];

}

generalizationView

{

source: a66434bc-303b-4f39-a04f-a748c47658c8;

target: 6e412d99-3996-4d4d-825a-14eddd935ca7;

points: [(659.8835815387845,728.2922107619345),(669.5333130664569,218.5154925680954)];

}

}

###Relational

Database store::SustainabilityStatisticsDatabase

(

Schema legend\_datastore\_gracehopper

(

Table carbon\_dioxide\_emissions

(

country\_iso\_code VARCHAR(10) PRIMARY KEY,

year INTEGER PRIMARY KEY,

co2\_including\_luc FLOAT,

co2\_including\_luc\_per\_capita FLOAT,

co2\_including\_luc\_growth\_abs FLOAT,

gas\_co2 FLOAT,

gas\_co2\_per\_capita FLOAT,

cumulative\_gas\_co2 FLOAT,

land\_use\_change\_co2 FLOAT,

land\_use\_change\_co2\_per\_capita FLOAT,

cumulative\_luc\_co2 FLOAT,

oil\_co2 FLOAT,

oil\_co2\_per\_capita FLOAT,

cumulative\_oil\_co2 FLOAT,

cement\_co2 FLOAT,

cement\_co2\_per\_capita FLOAT,

cumulative\_cement\_co2 FLOAT,

flaring\_co2 FLOAT,

flaring\_co2\_per\_capita FLOAT,

cumulative\_flaring\_co2 FLOAT,

coal\_co2 FLOAT,

coal\_co2\_per\_capita FLOAT,

cumulative\_coal\_co2 FLOAT,

other\_industry\_co2 FLOAT,

other\_co2\_per\_capita FLOAT,

cumulative\_other\_co2 FLOAT

)

Table country

(

country\_iso\_code VARCHAR(10) PRIMARY KEY,

name VARCHAR(100),

year INTEGER PRIMARY KEY,

population FLOAT,

gdp FLOAT

)

Table surface\_temperature\_change

(

country\_iso\_code VARCHAR(10) PRIMARY KEY,

year INTEGER PRIMARY KEY,

temperature\_change FLOAT

)

)

Join country\_emmissions(legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code = legend\_datastore\_gracehopper.country.country\_iso\_code

and legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year = legend\_datastore\_gracehopper.country.year)

Join emissions\_self\_join(legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code = {target}.country\_iso\_code

and legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year = {target}.year)

Join country\_surface\_temperature\_change(legend\_datastore\_gracehopper.surface\_temperature\_change.country\_iso\_code = legend\_datastore\_gracehopper.country.country\_iso\_code

and legend\_datastore\_gracehopper.surface\_temperature\_change.year = legend\_datastore\_gracehopper.country.year)

)

###Pure

Class {doc.doc = 'Represents carbon dioxide emissions that can be captured across various drivers.'} model::CarbonDioxideEmissions

{

country: model::Country[1];

year: Integer[1];

annualCO2Emissions: Float[0..1];

annualCO2EmissionsPerCapita: Float[0..1];

}

Class model::CementProductionBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class model::CoalProductionBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class {doc.doc = 'Geographic location with yearly population.'} model::Country

{

{doc.doc = 'Geographic location.'} name: String[1];

{doc.doc = 'ISO 3166-1 alpha-3, three-letter country codes.'} isoCode: String[1];

{doc.doc = 'Year of observation.'} year: Integer[1];

{doc.doc = 'Population of each country or region.'} population: Float[1];

surfaceTemperature: model::SurfaceTemperature[1];

}

Class model::FlaringBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class model::GasProductionBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class model::LandUseBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class model::OilProductionBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class model::OtherIndustryCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

}

Class {doc.doc = 'Annual total production-based emissions of carbon dioxide measured in million tonnes.'} model::ProductionBasedCarbonDioxideEmissions extends model::CarbonDioxideEmissions

{

{doc.doc = 'Annual total production-based emissions of carbon dioxide (CO\u2082).'} annualGrowthInCO2Emissions: Float[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from cement.'} cementProductionBasedCO2Emissions: model::CementProductionBasedCarbonDioxideEmissions[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from coal.'} coalProductionBasedCO2Emissions: model::CoalProductionBasedCarbonDioxideEmissions[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from flaring.'} flaringBasedCO2Emissions: model::FlaringBasedCarbonDioxideEmissions[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from gas.'} gasProductionBasedCO2Emissions: model::GasProductionBasedCarbonDioxideEmissions[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from oil.'} oilProductionBasedCO2Emissions: model::OilProductionBasedCarbonDioxideEmissions[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from land-use change.'} landUseBasedCO2Emissions: model::LandUseBasedCarbonDioxideEmissions[0..1];

{doc.doc = 'Annual production-based emissions of carbon dioxide (CO\u2082) from other industry source.'} otherIndustryBasedCO2Emissions: model::OtherIndustryCarbonDioxideEmissions[0..1];

countriesWithEmissionsRecordedIn2020() {$this.country->filter(

x|$x.year == 2020

).name}: String[\*];

}

Class model::SurfaceTemperature

{

country: model::Country[1];

year: Integer[1];

temperatureChange: Float[0..1];

}

Association model::Country\_ProductionBasedCarbonDioxideEmissions

{

country: model::Country[1];

carbonDioxideEmissions: model::ProductionBasedCarbonDioxideEmissions[1];

}

function z\_cheatsheet::queries::Query1\_PartOne\_NumberOfCountries(): meta::pure::tds::TabularDataSet[1]

{

model::ProductionBasedCarbonDioxideEmissions.all()->filter(

x|!($x.annualCO2Emissions->isEmpty())

)->groupBy(

[],

[

agg(

x|$x.country.name,

x|$x->distinct()->count()

)

],

['Country/Name']

)->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

function z\_cheatsheet::queries::Query1\_PartTwo\_ListOfCountries(): meta::pure::tds::TabularDataSet[1]

{

model::ProductionBasedCarbonDioxideEmissions.all()->filter(

x|!($x.annualCO2Emissions->isEmpty())

)->project(

[

x|$x.country.name

],

['Country/Name']

)->distinct()->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

function z\_cheatsheet::queries::Query2\_Top10CO2Emissions(): meta::pure::tds::TabularDataSet[1]

{

model::ProductionBasedCarbonDioxideEmissions.all()->filter(

x|!($x.annualCO2Emissions->isEmpty()) &&

($x.year == 2021)

)->project(

[

x|$x.country.name,

x|$x.annualCO2Emissions

],

[

'Country/Name',

'Annual CO 2 Emissions'

]

)->sort(

[

desc('Annual CO 2 Emissions')

]

)->take(10)->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

function z\_cheatsheet::queries::Query3\_Top10AverageCO2Emissions(): meta::pure::tds::TabularDataSet[1]

{

model::ProductionBasedCarbonDioxideEmissions.all()->filter(

x|!($x.annualCO2Emissions->isEmpty())

)->groupBy(

[

x|$x.country.name

],

[

agg(

x|$x.annualCO2Emissions,

x|$x->average()

)

],

[

'Country/Name',

'Annual CO 2 Emissions'

]

)->sort(

[

desc('Annual CO 2 Emissions')

]

)->take(10)->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

function z\_cheatsheet::queries::Query4\_SurfTempByCountry(): meta::pure::tds::TabularDataSet[1]

{

model::SurfaceTemperature.all()->filter(

x|($x.year == 2020) &&

!($x.temperatureChange->isEmpty())

)->project(

[

x|$x.country.name,

x|$x.temperatureChange

],

[

'Country/Name',

'Temperature Change'

]

)->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

function z\_cheatsheet::queries::Query5\_Top10TempChange(): meta::pure::tds::TabularDataSet[1]

{

model::SurfaceTemperature.all()->filter(

x|($x.year >= 2016) &&

(($x.year <= 2021) &&

!($x.temperatureChange->isEmpty()))

)->project(

[

x|$x.country.name,

x|$x.temperatureChange

],

[

'Country/Name',

'Temperature Change'

]

)->sort(

[

desc('Temperature Change')

]

)->take(10)->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

function z\_cheatsheet::queries::Query6\_SortByTempChangeCO2(): meta::pure::tds::TabularDataSet[1]

{

model::SurfaceTemperature.all()->filter(

x|($x.country.year >= 2001) &&

(!($x.temperatureChange->isEmpty()) &&

!($x.country.carbonDioxideEmissions.annualCO2Emissions->isEmpty()))

)->groupBy(

[

x|$x.country.name

],

[

agg(

x|$x.temperatureChange,

x|$x->average()

),

agg(

x|$x.country.carbonDioxideEmissions.annualCO2Emissions,

x|$x->average()

)

],

[

'Country/Name',

'Temperature Change',

'Country/Carbon Dioxide Emissions/Annual CO 2 Emissions'

]

)->from(

mapping::SustainabilityStatisticsMapping,

runtime::SustainabilityRuntime

)

}

###Mapping

Mapping mapping::SustainabilityStatisticsMapping

(

\*model::Country[country]: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.country.country\_iso\_code

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.country

name: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.country.name,

isoCode: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.country.country\_iso\_code,

population: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.country.population,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.country.year,

carbonDioxideEmissions[model\_ProductionBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

surfaceTemperature[model\_SurfaceTemperature]: [store::SustainabilityStatisticsDatabase]@country\_surface\_temperature\_change

}

\*model::OilProductionBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.oil\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.oil\_co2\_per\_capita,

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions

}

\*model::LandUseBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.land\_use\_change\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.land\_use\_change\_co2\_per\_capita,

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions

}

\*model::GasProductionBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.gas\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.gas\_co2\_per\_capita,

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

}

\*model::CementProductionBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.cement\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.cement\_co2\_per\_capita

}

\*model::CoalProductionBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.coal\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.coal\_co2\_per\_capita

}

\*model::FlaringBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.flaring\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.flaring\_co2\_per\_capita

}

\*model::OtherIndustryCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.other\_industry\_co2,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.other\_co2\_per\_capita

}

\*model::ProductionBasedCarbonDioxideEmissions: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions

country[country]: [store::SustainabilityStatisticsDatabase]@country\_emmissions,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.year,

annualCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.co2\_including\_luc,

annualCO2EmissionsPerCapita: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.co2\_including\_luc\_per\_capita,

annualGrowthInCO2Emissions: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.carbon\_dioxide\_emissions.co2\_including\_luc\_growth\_abs,

cementProductionBasedCO2Emissions[model\_CementProductionBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join,

coalProductionBasedCO2Emissions[model\_CoalProductionBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join,

flaringBasedCO2Emissions[model\_FlaringBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join,

gasProductionBasedCO2Emissions[model\_GasProductionBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join,

oilProductionBasedCO2Emissions[model\_OilProductionBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join,

landUseBasedCO2Emissions[model\_LandUseBasedCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join,

otherIndustryBasedCO2Emissions[model\_OtherIndustryCarbonDioxideEmissions]: [store::SustainabilityStatisticsDatabase]@emissions\_self\_join

}

\*model::SurfaceTemperature: Relational

{

~primaryKey

(

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.surface\_temperature\_change.country\_iso\_code,

[store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.surface\_temperature\_change.year

)

~mainTable [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.surface\_temperature\_change

country[country]: [store::SustainabilityStatisticsDatabase]@country\_surface\_temperature\_change,

year: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.surface\_temperature\_change.year,

temperatureChange: [store::SustainabilityStatisticsDatabase]legend\_datastore\_gracehopper.surface\_temperature\_change.temperature\_change

}

MappingTests

[

test\_1

(

query: |model::Country.all()->project(

[

x|$x.isoCode,

x|$x.name,

x|$x.year,

x|$x.population

],

[

'Iso Code',

'Name',

'Year',

'Population'

]

);

data:

[

<Relational, CSV, store::SustainabilityStatisticsDatabase,

'legend\_datastore\_gracehopper\n'+

'country\n'+

'country\_iso\_code,name,year,population,gdp\n'+

'AFG,Afghanistan,2010,3752993,99999\n'+

'\n\n\n'

>

];

assert: '[{"values":["AFG","Afghanistan",2010,3752993.0]}]';

),

test\_3

(

query: |model::ProductionBasedCarbonDioxideEmissions.all()->filter(

x|$x.year == 2000

)->project(

[

x|$x.country.name,

x|$x.year,

x|$x.annualCO2Emissions,

x|$x.annualCO2EmissionsPerCapita,

x|$x.annualGrowthInCO2Emissions,

x|$x.cementProductionBasedCO2Emissions.annualCO2Emissions,

x|$x.cementProductionBasedCO2Emissions.annualCO2EmissionsPerCapita,

x|$x.coalProductionBasedCO2Emissions.annualCO2Emissions,

x|$x.coalProductionBasedCO2Emissions.annualCO2EmissionsPerCapita,

x|$x.flaringBasedCO2Emissions.annualCO2Emissions,

x|$x.flaringBasedCO2Emissions.annualCO2EmissionsPerCapita,

x|$x.gasProductionBasedCO2Emissions.annualCO2Emissions,

x|$x.gasProductionBasedCO2Emissions.annualCO2EmissionsPerCapita,

x|$x.landUseBasedCO2Emissions.annualCO2Emissions,

x|$x.landUseBasedCO2Emissions.annualCO2EmissionsPerCapita,

x|$x.oilProductionBasedCO2Emissions.annualCO2Emissions,

x|$x.oilProductionBasedCO2Emissions.annualCO2EmissionsPerCapita,

x|$x.otherIndustryBasedCO2Emissions.annualCO2Emissions,

x|$x.otherIndustryBasedCO2Emissions.annualCO2EmissionsPerCapita

],

[

'Country/Name',

'Year',

'Annual CO 2 Emissions',

'Annual CO 2 Emissions Per Capita',

'Annual Growth In CO 2 Emissions',

'Cement Production Based CO 2 Emissions/Annual CO 2 Emissions',

'Cement Production Based CO 2 Emissions/Annual CO 2 Emissions Per Capita',

'Coal Production Based CO 2 Emissions/Annual CO 2 Emissions',

'Coal Production Based CO 2 Emissions/Annual CO 2 Emissions Per Capita',

'Flaring Based CO 2 Emissions/Annual CO 2 Emissions',

'Flaring Based CO 2 Emissions/Annual CO 2 Emissions Per Capita',

'Gas Production Based CO 2 Emissions/Annual CO 2 Emissions',

'Gas Production Based CO 2 Emissions/Annual CO 2 Emissions Per Capita',

'Land Use Based CO 2 Emissions/Annual CO 2 Emissions',

'Land Use Based CO 2 Emissions/Annual CO 2 Emissions Per Capita',

'Oil Production Based CO 2 Emissions/Annual CO 2 Emissions',

'Oil Production Based CO 2 Emissions/Annual CO 2 Emissions Per Capita',

'Other Industry Based CO 2 Emissions/Annual CO 2 Emissions',

'Other Industry Based CO 2 Emissions/Annual CO 2 Emissions Per Capita'

]

);

data:

[

<Relational, CSV, store::SustainabilityStatisticsDatabase,

'legend\_datastore\_gracehopper\n'+

'country\n'+

'country\_iso\_code,name,year,population,gdp\n'+

'AFG,Afghanistan,2000,3752993,99999\n'+

'-----\n'+

'legend\_datastore\_gracehopper\n'+

'carbon\_dioxide\_emissions\n'+

'year,country\_iso\_code,co2\_including\_luc,co2\_including\_luc\_per\_capita,co2\_including\_luc\_growth\_abs,cement\_co2,cement\_co2\_per\_capita,cumulative\_cement\_co2,coal\_co2,coal\_co2\_per\_capita,cumulative\_coal\_co2,flaring\_co2,flaring\_co2\_per\_capita,cumulative\_flaring\_co2,gas\_co2,gas\_co2\_per\_capita,cumulative\_gas\_co2,land\_use\_change\_co2,land\_use\_change\_co2\_per\_capita,cumulative\_luc\_co2,oil\_co2,oil\_co2\_per\_capita,cumulative\_oil\_co2,other\_industry\_co2,other\_co2\_per\_capita,cumulative\_other\_co2\n'+

'2000,AFG,0.864,0.044,-0.154,0.01,0.001,1.969,0.004,0,11.851,0.022,0.001,5.934,0.224,0.011,13.897,-0.183,-0.009,836.015,0.788,0.04,38.798,,,\n'+

'2001,AFG,1.948,0.099,1.085,0.007,0,1.975,0.07,0.004,11.921,0.022,0.001,5.956,0.209,0.011,14.106,0.879,0.045,836.894,0.762,0.039,39.56,,,\n'+

'\n\n\n'

>

];

assert: '[{"values":["Afghanistan",2000,0.864,0.044,-0.154,0.01,0.001,0.004,0.0,0.022,0.001,0.224,0.011,-0.183,-0.009,0.788,0.04,null,null]}]';

)

]

)

###Connection

RelationalDatabaseConnection connection::SustainabilityPostgresConnection

{

store: store::SustainabilityStatisticsDatabase;

type: Postgres;

quoteIdentifiers: false;

specification: Static

{

name: 'legend';

host: 'localhost';

port: 5432;

};

auth: UserNamePassword

{

userNameVaultReference: 'postgres.username';

passwordVaultReference: 'postgres.password';

};

}

###Runtime

Runtime runtime::SustainabilityRuntime

{

mappings:

[

mapping::SustainabilityStatisticsMapping

];

connections:

[

store::SustainabilityStatisticsDatabase:

[

connection\_1: connection::SustainabilityPostgresConnection

]

];

}