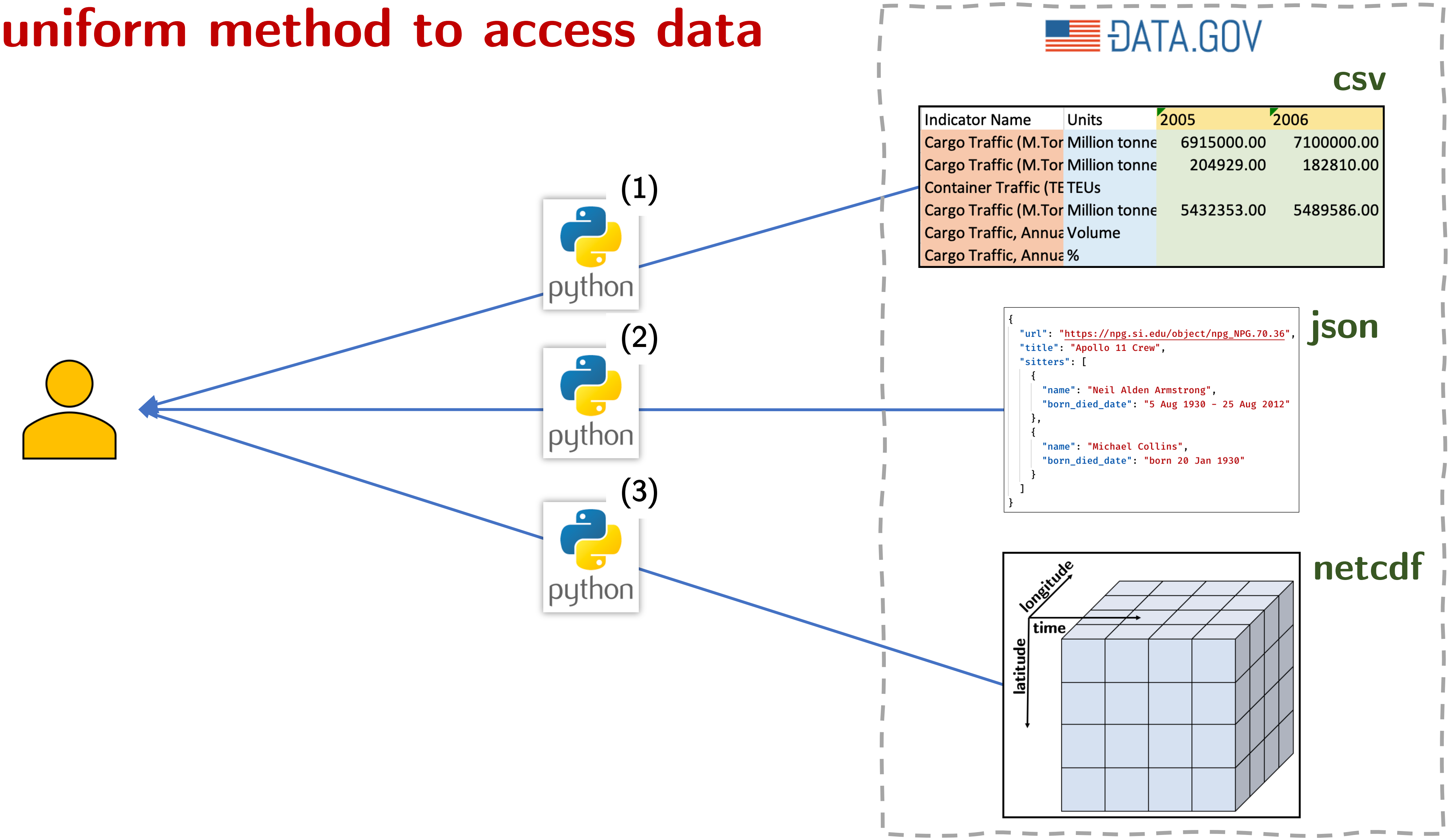
# D-REPR: A Language For Describing And Mapping Diversely-Structured Data Sources To RDF

Binh Vu, Jay Pujara, and Craig Knoblock

# Motivating example

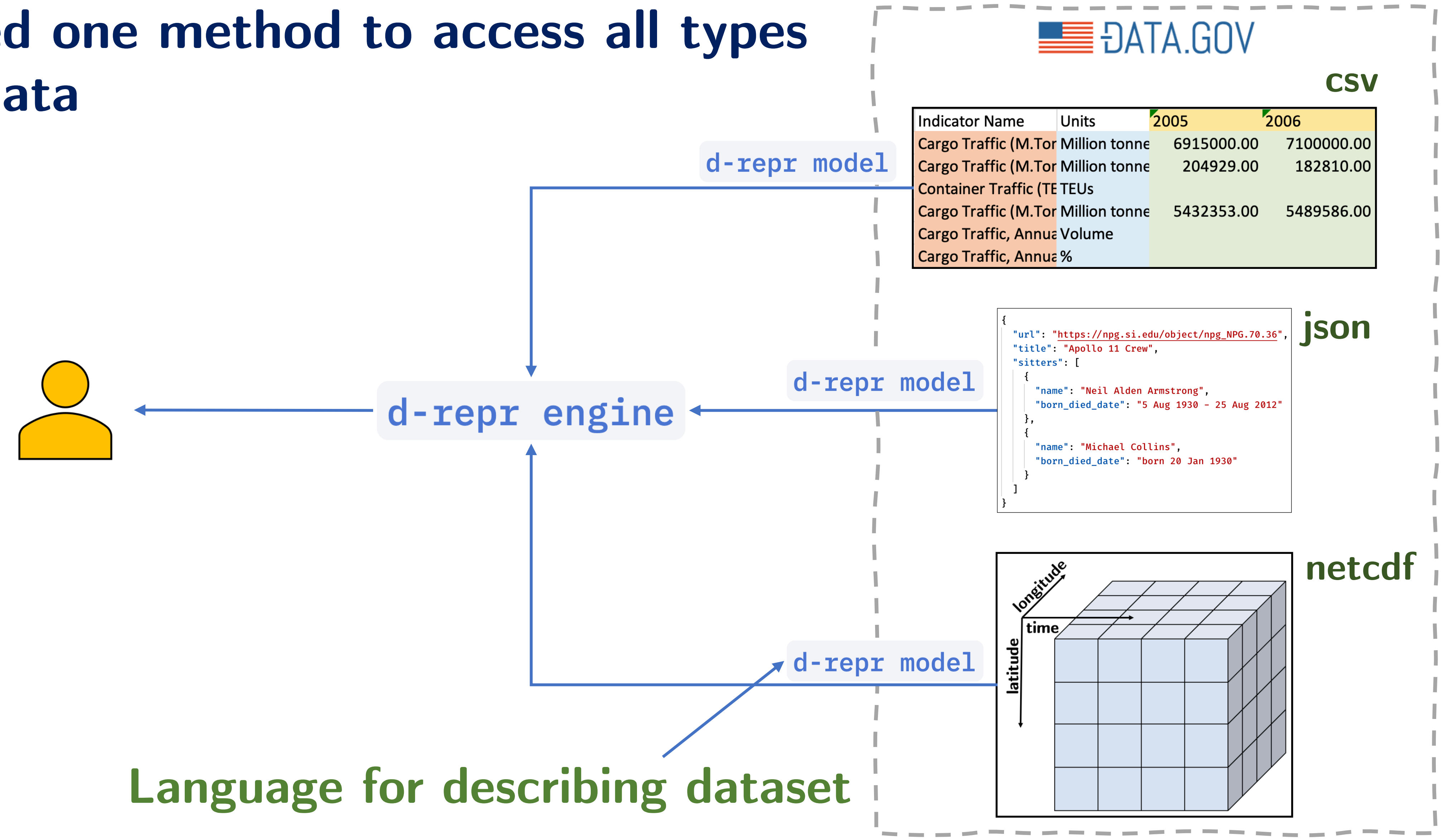
**No uniform method to access data**




| Indicator Name | Units | 2005 | 2006 |
|---|---|---|---|
| Cargo Traffic (M.Tor | Million tonne | 6915000.00 | 7100000.00 |
| Cargo Traffic (M.Tor | Million tonne | 204929.00 | 182810.00 |
| Container Traffic (TE | TEUs | | |
| Cargo Traffic (M.Tor | Million tonne | 5432353.00 | 5489586.00 |
| Cargo Traffic, Annua | Volume | | |
| Cargo Traffic, Annua | % | | |

# Motivating example

## Need one method to access all types of data




| Indicator Name | Units | 2005 | 2006 |
|---|---|---|---|
| Cargo Traffic (M.Tor | Million tonne | 6915000.00 | 7100000.00 |
| Cargo Traffic (M.Tor | Million tonne | 204929.00 | 182810.00 |
| Container Traffic (TE | TEUs | | |
| Cargo Traffic (M.Tor | Million tonne | 5432353.00 | 5489586.00 |
| Cargo Traffic, Annua | Volume | | |
| Cargo Traffic, Annua | % | | |

# Heterogeneous datasets

- Multiple **formats**: CSV, JSON, XLSX, NetCDF4, …

| | | 2016 | |
|---|---|---|---|
| Indicator | Age Group | Male | Female |
| LIFE_0035 | <1 year | 57.7 | 59.6 |
| LIFE_0035 | 1-4 years | 60.6 | 62.1 |

```
,,2016,
Indicator,Age Group,Male,Female
LIFE_0035,<1 year,57.7,59.6
LIFE_0035,1-4 years,60.6,62.1
```

```
[
    {
        "indicator": "LIFE_0035",
        "age group": "< 1 year",
        "gender": "male",
        "year": "2016",
        "value": 57.7
    },
    {
        "indicator": "LIFE_0035",
        "age group": "< 1 year",
        "gender": "female",
        "year": "2016",
        "value": 59.6
    }
]
```

```
<obs>
    <ob>
        <indicator>LIFE_0035</indicator>
        <age_group>&lt;1 year</age_group>
        <gender>male</gender>
        <year>2016</year>
        <value>57.7</value>
    </ob>
    <ob>
        <indicator>LIFE_0035</indicator>
        <age_group>&lt;1 year</age_group>
        <gender>female</gender>
        <year>2016</year>
        <value>59.6</value>
    </ob>
</obs>
```

# Heterogeneous datasets

- Same format, multiple **layouts**

| Indicator | Age Group | Gender | Year | Value |
|---|---|---|---|---|
| LIFE_0035 | < 1 year | Male | 2016 | 57.7 |
| LIFE_0035 | < 1 year | Female | 2016 | 59.6 |
| LIFE_0035 | 1-4 years | Male | 2016 | 60.6 |
| LIFE_0035 | 1-4 years | Female | 2016 | 62.1 |

| LIFE_0035 | | |
|---|---|---|
| Age Group | Gender | Observation |
| 2016 | | |
| < 1 year | Male | 59.6 |
| 1-4 years | Female | 62.1 |

| | | 2016 | |
|---|---|---|---|
| Indicator | Age Group | Male | Female |
| LIFE_0035 | < 1 year | 57.7 | 59.6 |
| LIFE_0035 | 1-4 years | 60.6 | 62.1 |

| | 2016 | |
|---|---|---|
| Age Group | Male | Female |
| < 1 year | 57.7 | 59.6 |
| 1-4 years | 60.6 | 62.1 |

LIFE_0035 LIFE_0029 +

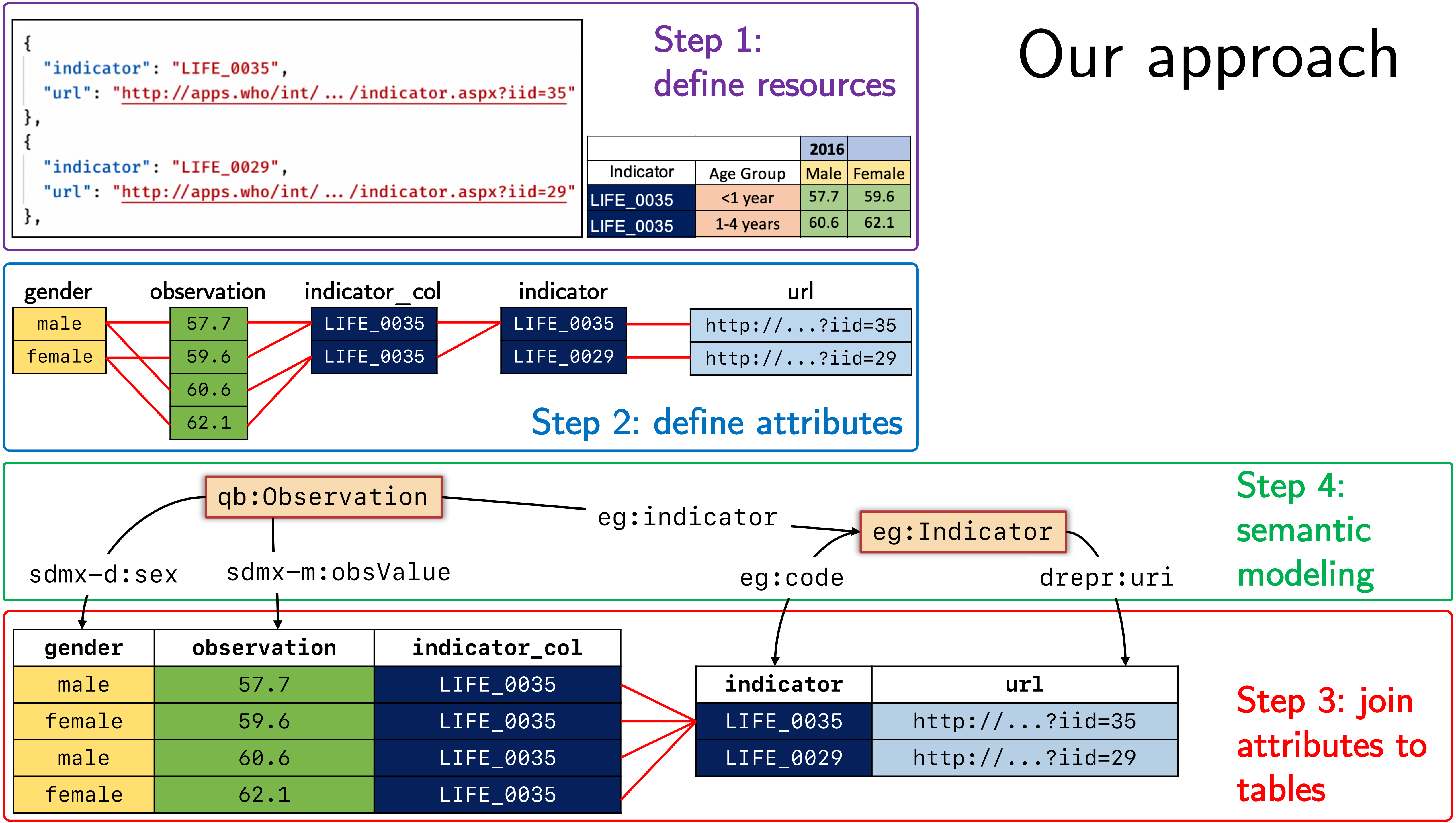
# Related work

- Mapping nested relational datasets:
  - RML (Dimou et al, 2014), KR2RML (Slepicka et al 2015), xR2RML (Michel et al, 2015), etc.
  - Can handle multiple formats but only work for nested relational model layout
- Mapping tabular datasets:
  - XLWrap (Langegger et al, 2009), M2 (O'Connor et al, 2010), T2WML (Szekely et al, 2019)
  - Can handle multiple layouts, but support only tabular formats

# Contributions

- A generic language to easily for describing and mapping heterogeneous datasets to RDF
  - It's capable of mapping wide variety of data sources and goes beyond the set of sources that existing languages support.
- The language is extensible to new formats and layouts
- An efficient engine to convert datasets to RDF

# Our approach



# Step 1: Resources

- A resource can be a physical file, SQL table, etc.
- Syntax:

```
resources:
    <resource_id>:
        type: <resource_type>
> preprocessing: ...
> attributes: ...
> alignments: ...
> semantic_model: ...
```

- Example:

```
resources:
    life_tbl:
        type: csv
    indicators:
        type: json
```

life_table.csv

| | | 2016 | |
|---|---|---|---|
| Indicator | Age Group | Male | Female |
| LIFE_0035 | <1 year | 57.7 | 59.6 |
| LIFE_0035 | 1-4 years | 60.6 | 62.1 |

indicators.json

```
{
  "indicator": "LIFE_0035",
  "url": "http://apps.who/int/ ... /indicator.aspx?iid=35"
},
{
  "indicator": "LIFE_0029",
  "url": "http://apps.who/int/ ... /indicator.aspx?iid=29"
},
```

# Step 2: Attributes

- Containing values that belong to a group
- Syntax

```
> resources: ⋯
> preprocessing: ⋯
  attributes:
      <attribute_id>:
          [resource_id]: <resource_id>
          path: <json_path>
          [unique]: false
          [missing_values]: [<value_0>, <value_1>, … ]
> alignments: ⋯
> semantic_model: ⋯
```

life_table.csv

| | | 2016 | |
|---|---|---|---|
| Indicator | Age Group | Male | Female |
| LIFE_0035 | <1 year | 57.7 | 59.6 |
| LIFE_0035 | 1-4 years | 60.6 | 62.1 |

```
  attributes:
      year:
          resource_id: life_tbl
          path: $[0][2:]
      gender:
          resource_id: life_tbl
          path: $[1][2:]
      indicator_col:
          resource_id: life_tbl
          path: $[2:][0]
      age_group:
          resource_id: life_tbl
          path: $[2:][1]
      observation:
          resource_id: life_tbl
          path: $[2:][2:]
>     indicator: ⋯
>     url: ⋯
```

# Step 2: Attributes

**indicators.json**

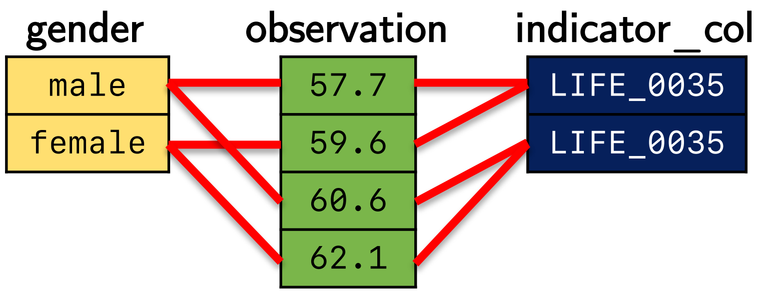
```
{
  "indicator": "LIFE_0035",
  "url": "http://apps.who/int/ … /indicator.aspx?iid=35"
},
{
  "indicator": "LIFE_0029",
  "url": "http://apps.who/int/ … /indicator.aspx?iid=29"
},
```

```
attributes:
    year: …
    gender: …
    indicator_col: …
    age_group: …
    observation: …
    indicator:
        resource_id: indicators
        path: $[:].indicator
        unique: true
    url:
        resource_id: indicators
        path: $[:].url
        unique: true
```

# Step 3: Alignments

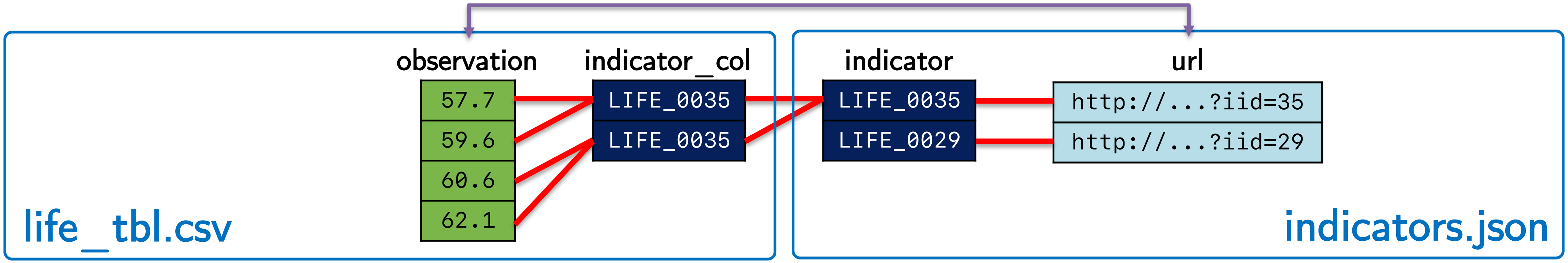
- Explicitly specifying the layout through alignments

| Indicator | male | female |
|---|---|---|
| LIFE_0035 | 57.7 | 59.6 |
| LIFE_0035 | 60.6 | 62.1 |




| gender | observation | indicator_col |
|---|---|---|
| male | 57.7 | LIFE_0035 |
| female | 59.6 | LIFE_0035 |
| male | 60.6 | LIFE_0035 |
| female | 62.1 | LIFE_0035 |

- For linking across resources



# Step 3: Alignments

- Join by value (equi-join)

```
{
  "indicator": "LIFE_0035",
  "url": "http://apps.who/int/ … /indicator.aspx?iid=35"
},
{
  "indicator": "LIFE_0029",
  "url": "http://apps.who/int/ … /indicator.aspx?iid=29"
},
```

| | | 2016 | |
|---|---|---|---|
| Indicator | Age Group | Male | Female |
| LIFE_0035 | <1 year | 57.7 | 59.6 |
| LIFE_0035 | 1-4 years | 60.6 | 62.1 |

- Syntax

```
> resources: …
> preprocessing: …
> attributes: …
  alignments:
      - type: <join_type>
        source: <attribute_id>
        target: <attribute_id>
        # ..optional arguments depends on the alignment type..
> semantic_model: …
```

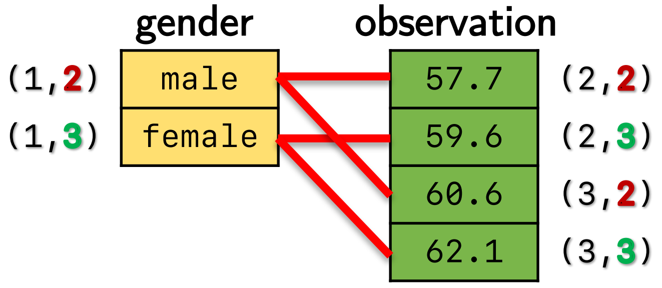
```
alignments:
    - type: value
      source: indicator_col
      target: indicator
```

# Step 3: Alignments

- Join by positions in the dataset

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | 2016 | |
| 1 | Indicator | Age Group | male (1,2) | female (1,3) |
| 2 | LIFE_0035 | <1 year | 57.7 (2,2) | 59.6 (2,3) |
| 3 | LIFE_0035 | 1–4 years | 60.6 (3,2) | 62.1 (3,3) |

dimension 0 (row) dimension 1 (column)



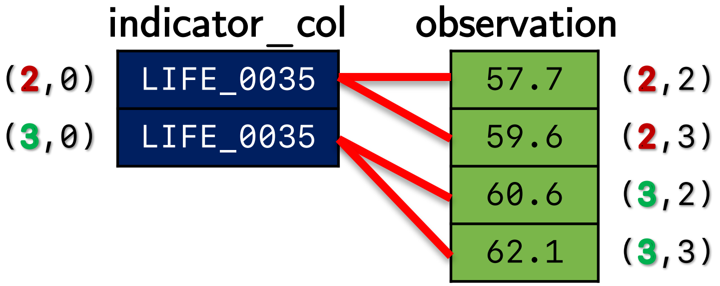

```
alignments:
  - type: value…
  - type: dimension
    source: observation
    target: gender
    aligned_dims:
      - source: 1
        target: 1
```

# Step 3: Alignments

- Join by positions in the dataset

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | 2016 | |
| 1 | Indicator | Age Group | male | female |
| 2 | LIFE_0035 (2,0) | <1 year | 57.7 (2,2) | 59.6 (2,3) |
| 3 | LIFE_0035 (3,0) | 1–4 years | 60.6 (3,2) | 62.1 (3,3) |

dimension 0 (row)    dimension 1 (column)




```yaml
alignments:
  - type: value ...
  - type: dimension ...
  - type: dimension
    source: observation
    target: indicator_col
    aligned_dims:
      - source: 0
        target: 0
```

# Step 3: Alignments

- Join by positions in the dataset

```
[{
    "departments": {
      "people": [{
          "name": "Peter",
          "phone": "213-266-2777"
        },
        {
          "name": "John",
          "phone": "222-222-2222"
        } /* more */]
    }
  } /* more */]
```

Sample data

dimension 0 dimension 1 dimension 2 dimension 3 dimension 4

```
Name:  $.*.departments.people.*.name
Phone: $.*.departments.people.*.phone
```

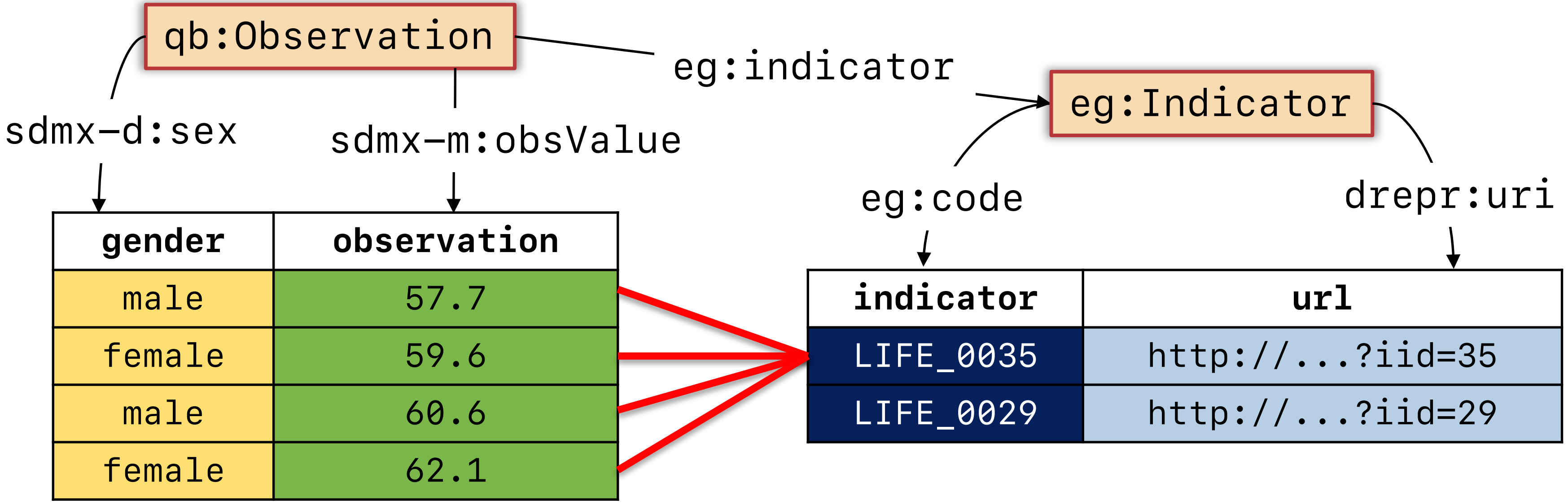
Aligned in dimensions 0 and 3

# Step 3: Alignments

- Easy to incorporate new alignment function
- Users only need to define the minimum number of joins (N-1) because the engine can infer the rest via composition.

# Step 4: Semantic Model

- Using ontologies to describe your data (classes and predicates)

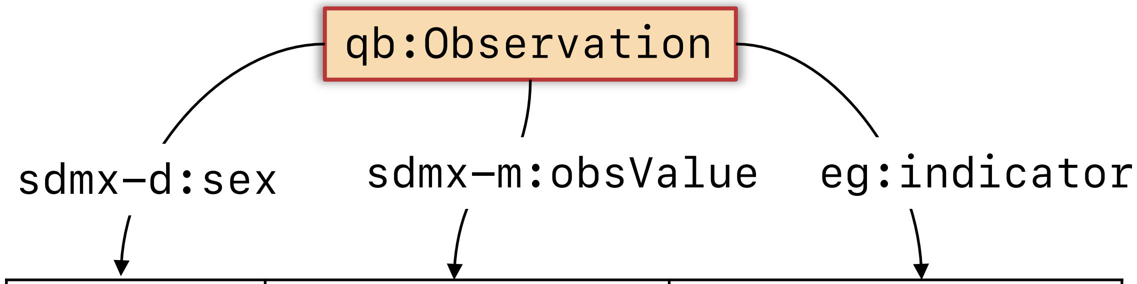



- Syntax

```
semantic_model:
    data_nodes:
        observation: qb:Observation:1--sdmx-m:obsValue
        year: qb:Observation:1--sdmx-d:refPeriod
        indicator: eg:Indicator:1--eg:code
        url: eg:Indicator:1--drepr:uri
    relations:
        - qb:Observation:1--eg:indicator--eg:Indicator:1
```

# Step 4: Semantic Model

- Users can create arbitrary semantic model, even when using attributes across multiple resources




| gender | observation | url |
|---|---|---|
| male | 57.7 | http://...?iid=35 |
| female | 59.6 | http://...?iid=35 |
| male | 60.6 | http://...?iid=35 |
| female | 62.1 | http://...?iid=35 |

# Data cleaning (optional)

- Users can write python function to clean or transform the data

```
preprocessing:
  - type: pmap
    input:
      resource: life_tbl
      path: $[0][2:]
    code: |
      if value == "":
        return context.get_left_value(index)
      return value
```

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | 2016 | |
| 1 | Indicator | Age Group | male | |
| 2 | LIFE_0035 | <1 year | 57.7 | 59.6 |
| 3 | LIFE_0035 | 1–4 years | 60.6 | 62.1 |

index = (0,3)
value = ""

- Can re-use functions or existing libraries

# Evaluation

- Coverage of D-REPR
  - Randomly sampling 700 datasets from data.gov
  - Modeling datasets of different formats and layouts

a. Children and Family Health

```
{
  "columns": [
    {"name": "teenbir10", "description": "Teen Birth Rate ... (2010)"},
    {"name": "teenbir11", "description": "Teen Birth Rate ... (2011)"},
    ... ],
  "data": [
    [ ... , "Allendale/Irvington/S. Hilton",   "55.0", "58.1",  ... ],
    [ ... , "Beechfield/Ten Hills/West Hills", "42.8", "21.4",  ... ],
    ... ],
  ...
}
```

b. Sugar production by sugar beet and sugarcane processors

| FY 2008 | JAN | FEB | MAR | APR | MAY | JUN |
|---|---|---|---|---|---|---|
| From domestic sugar beets | 661,586 | 485,126 | 423,775 | 337,473 | 216,526 | 82,987 |
| From imported sugar beets | 0 | 37,160 | 0 | 0 | 0 | 0 |
| **Subtotal** | 661,586 | 522,287 | 423,775 | 337,473 | 216,526 | 82,987 |
| Cane production: | | | | | | |
| Florida | 321,414 | 253,438 | 242,560 | 92,302 | 47,237 | 0 |
| ... | ... | | | | | |
| **Subtotal** | 378,919 | 283,190 | 289,237 | 108,826 | 68,504 | 30,903 |
| **Total** | 1,040,505 | 805,476 | 713,012 | 446,298 | 285,030 | 113,889 |

**Cannot be modeled with Nested Relational Models!**

# Evaluation

- Runtime of D-REPR engine (ms)
  - Mapping large CSV files (row-based table) containing (name, phone, address)
  - Generating 1.3m triples / second (15 times faster than KR2RML)

| | Number of records | | | | |
|---|---|---|---|---|---|
| Tools | 5,000 | 10,000 | 20,000 | 40,000 | 80,000 |
| D-REPR | 33.44 | 69.84 | 132.00 | 267.50 | 551.24 |
| KR2RML | 1368.00 | 1776.33 | 3276.66 | 4990.33 | 8305.33 |
| Morph | 4812.00 | 14949.66 | 65961.33 | - | - |

# Discussion and Future work

- A novel **generic** data representation language: D-REPR
  - Uses a declarative approach
  - Works for heterogeneous datasets of different formats and layouts
- Open source: https://github.com/usc-isi-i2/d-repr
- Future work:
  - (Semi-)automatically generating D-REPR models
  - UI for annotating datasets
  - Improving efficiency of D-REPR's engine by doing parallel processing

# References


- [1] RML: Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. 2014. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data
- [2] KR2RML: Jason Slepicka, Chengye Yin, Pedro Szekely, and Craig A. Knoblock. 2015. KR2RML: An Alternative Interpretation of R2RML for Heterogenous Sources
- [3] xR2RML: Franck Michel, Loïc Djimenou, Catherine Faron Zucker, and Johan Montagnat. 2015. Translation of relational and non-relational databases into RDF with xR2RML
- [4] XLWrap: Andreas Langegger and Wolfram Wöß. 2009. XLWrap — Querying and Inte- grating Arbitrary Spreadsheets with SPARQL
- [5] Martin J O'Connor, Christian Halaschek-Wiener, and Mark A Musen. 2010. M2: A Language for Mapping Spreadsheets to OWL