How to convert a csv to an Xarray with chosen dimension ?

1 - CSV data -> llist data

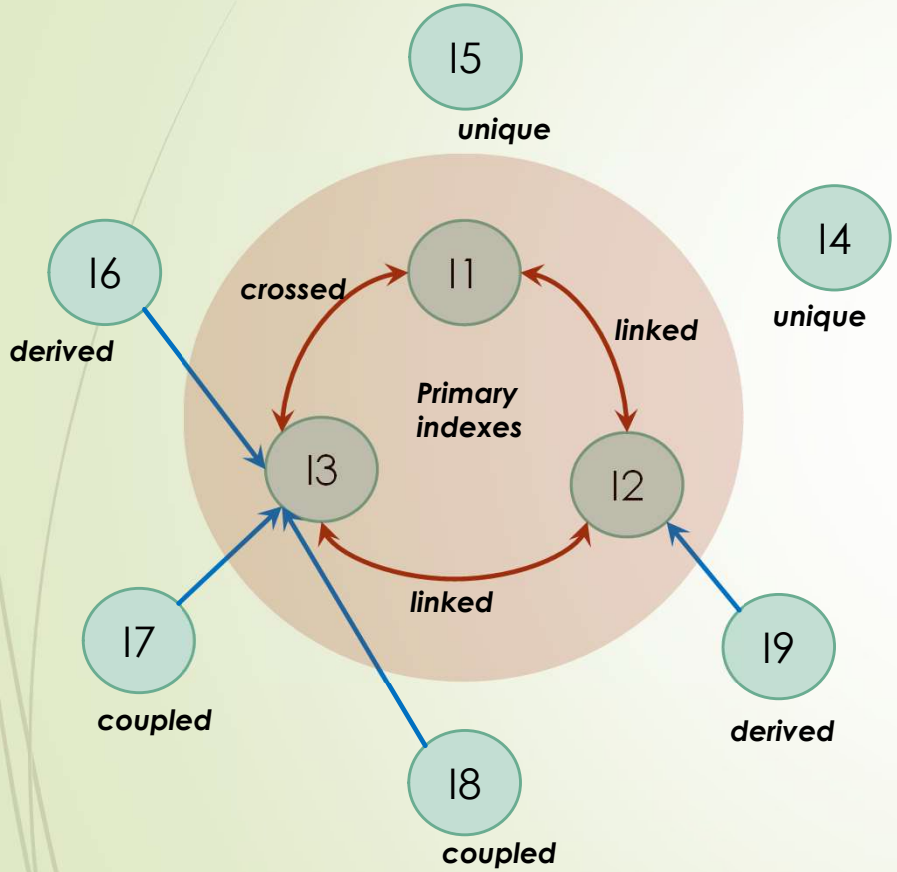
Example :

Name	City	Age	Subject	score
Paul	Paris	16	Math	10
Lea	Paris	15	Math	12
Lea	Paris	15	English	15

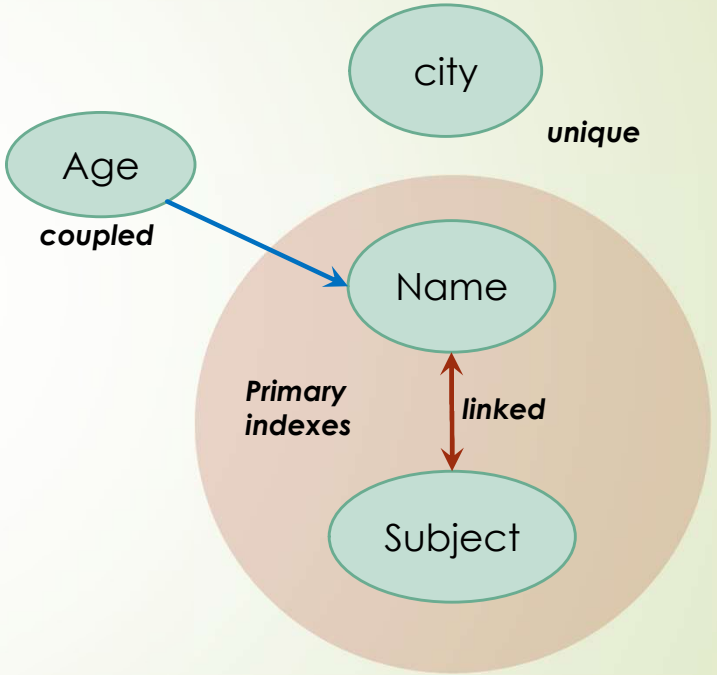
iindex

Name	City	Age	Subject	result: score
1	0	0	1	10
0	0	1	1	12
0	0	1	0	15

2 - Ilist -> Canonical format

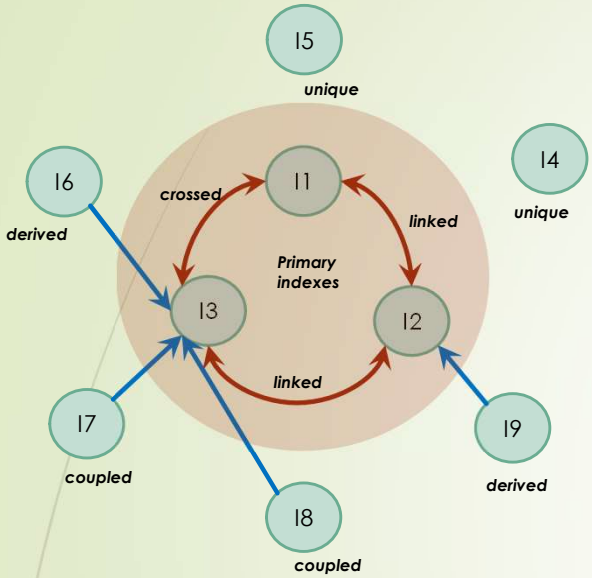


Example :

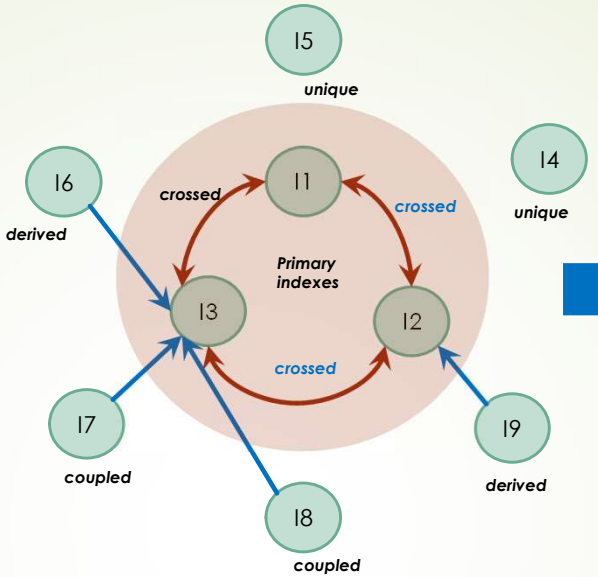


Explanation : See Appendix

3 - Xarray generation

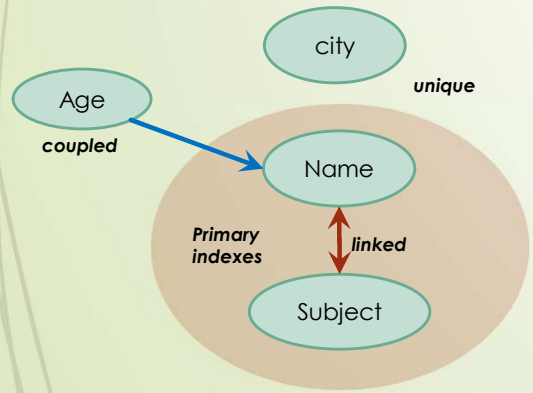


Full



- Primary indexes
  -> **Xarray dims**
- Derived/coupled indexes
  -> **Xarray coords**
- Indexed value
  -> **Xarray data**
- Unique index
  -> **Xarray attrs**

Example :




```
In [353]: il = Ilist.Iedic({'score'   : [10,12,15]},
     ...:                  {'name'    : ['Paul', 'Lea', 'Lea'],
     ...:                   'city'    : ['Paris', 'Paris', 'Paris'],
     ...:                   'age'     : [16,15,15],
     ...:                   'subject' : ['math', 'math', 'english']})

In [354]: il.to_xarray(fillvalue=math.nan)
Out[354]:
<xarray.DataArray 'score' (name: 2, subject: 2)>
array([[15., 12.],
       [nan, 10.]])
Coordinates:
  * name     (name) <U4 'Lea' 'Paul'
    age      (name) int32 15 16
  * subject  (subject) <U7 'english' 'math'
Attributes:
    city:     Paris
```

Appendix – Indexed List

0 – Ilist Principles

1 - Index analysis

2 - Matrix generation

3 - Aggregation

4 – Format, storage

https://loco-philippe.github.io/ES/ilist.html

0 - Ilist (Indexed list)

List of values :

\+ Age : [12, 28, 39, 58]

List of indexes :

Name : [Paul, John, Lea, Cat]

City : [Paris, Metz, Rennes, Bollène]

....

Name	city	Age
Paul	Paris	12
John	Metz	28
Lea	Rennes	39
Cat	Bollène	58

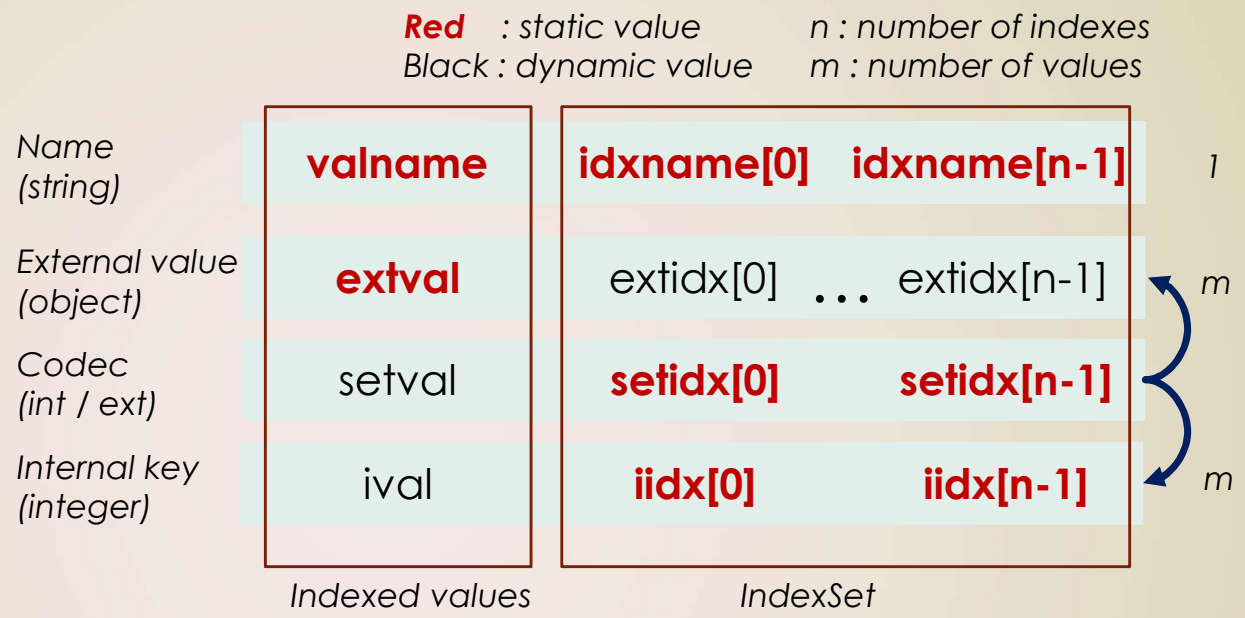
Example : csv file, measurement, log

Note : indexed values and index values can be every kind of object

0 – Data structure

Two levels

- **External values**
- **Internal keys** (no duplication)



Example

External value

score
name
age
subject

10	12	15
Paul	Lea	Lea
16	15	15
math	math	english

val ⟷ idx

Internal key

0	1	2
1	0	0
0	1	1
1	1	0

valname, idxname — *extval, extidx* — *ival, iidx*

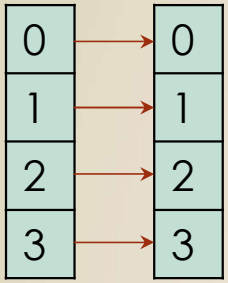
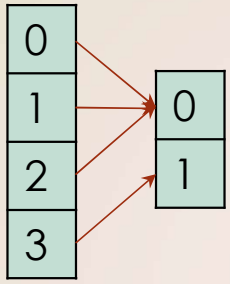
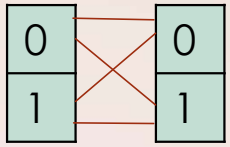
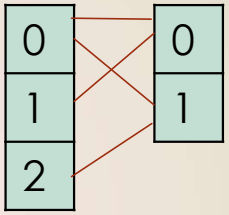
1 - Index categories

External Value	v	**[Anne, Paul, John]**	**[Anne, Anne, Anne]**	**[Anne, Paul, Anne]**
Internal key	i	0 1 2	0	0 1
Type		*complete*	*unique*	*mixte*
Property		Rate : 1 Disttomax : 0	Rate : 0 Disttomin : 0	0 < Rate < 1 m < dist < M

M = len(v)
m = 1
x = len(i)

Rate : (M – x) / (M – m)
Dist to min : x - m
Dist to max : M - x

1 - linking categories

External value v1	[Anne, Paul, John, Lea]	[Anne, Paul, John, Lea]	[Anne, Paul, Anne, Paul]	[Anne, Paul, Anne, Lea]
v2	[25, 26, 15, 35]	[25, 25, 25, 12]	[25, 25, 12, 12]	[25, 25, 12, 12]
Internal key i1 i2				
Type	coupled (asymmetrical)	derived (asymmetrical)	crossed	linked
Property	Rate : 0 Disttomin : 0	Rate : 0 Disttomin : 0	Rate : 1 Disttomax : 0	0 < Rate < 1 m < dist < M

M = len(i1) * len(i2)
m = max(len(i1), len(i2)
x = len(index(v1, v2))

Rate : (M – x) / (M – m)
Dist to min : x - m
Dist to max : M - x

- **Properties**
  - If one index is complete, all the indexes are derived from it
  - If one index is unique, it is derived from all other indexes
  - If A is derived (coupled) from B and B is derived (coupled) from C, A is derived (coupled) from C
  - If A is coupled from B, all the relationships with other indexes are identical

1 - Global properties

- **IndexSet**
  - Set of index with the same value lenght

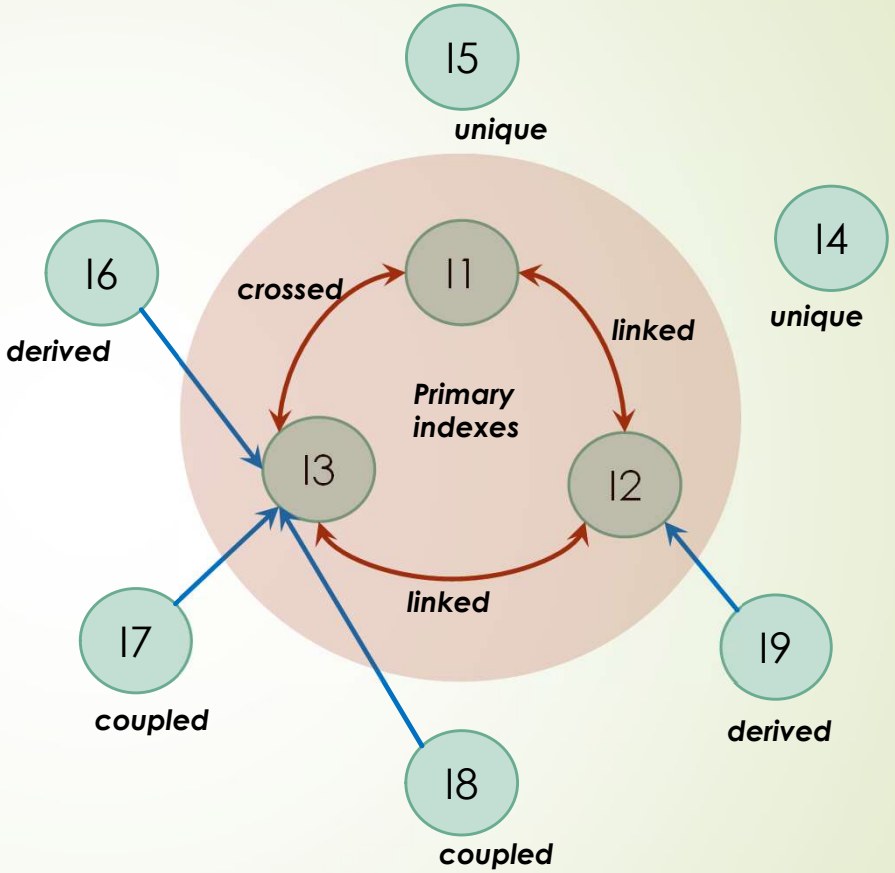
- **Index definition**
  - An index is derived if it's derived from at least one other index
  - An index is coupled if it's coupled from at least one other index
  - An Index is primary if it's not coupled, not derived and not unique

- **Indexset definition**
  - Dimension : number of primary indexes
  - Complete : An indexSet is complete if all the non coupled indexes are crossed with each other non coupled index
  - Full : An indexSet is full if all the primary indexes are crossed with each other primary index

- **Properties**
  - **A derived or coupled index is derived or coupled from a single primary index**
  - **The number of values of a full indexset is the product of the primary indexes length**
  - **A full indexSet is complete**
  - **A full IndexSet can be transformed in a Matrix with the dimension of the indexset**
  - **A complete Indexset can be expressed in a flat list of values (with order)**

1 – Canonical format

- **Primary indexes**
  - Linked or crossed whit each other
- **Derived or coupled indexes**
  - Associated with a single primary index
- **Unique indexes**
  - Not associated



1 - Example

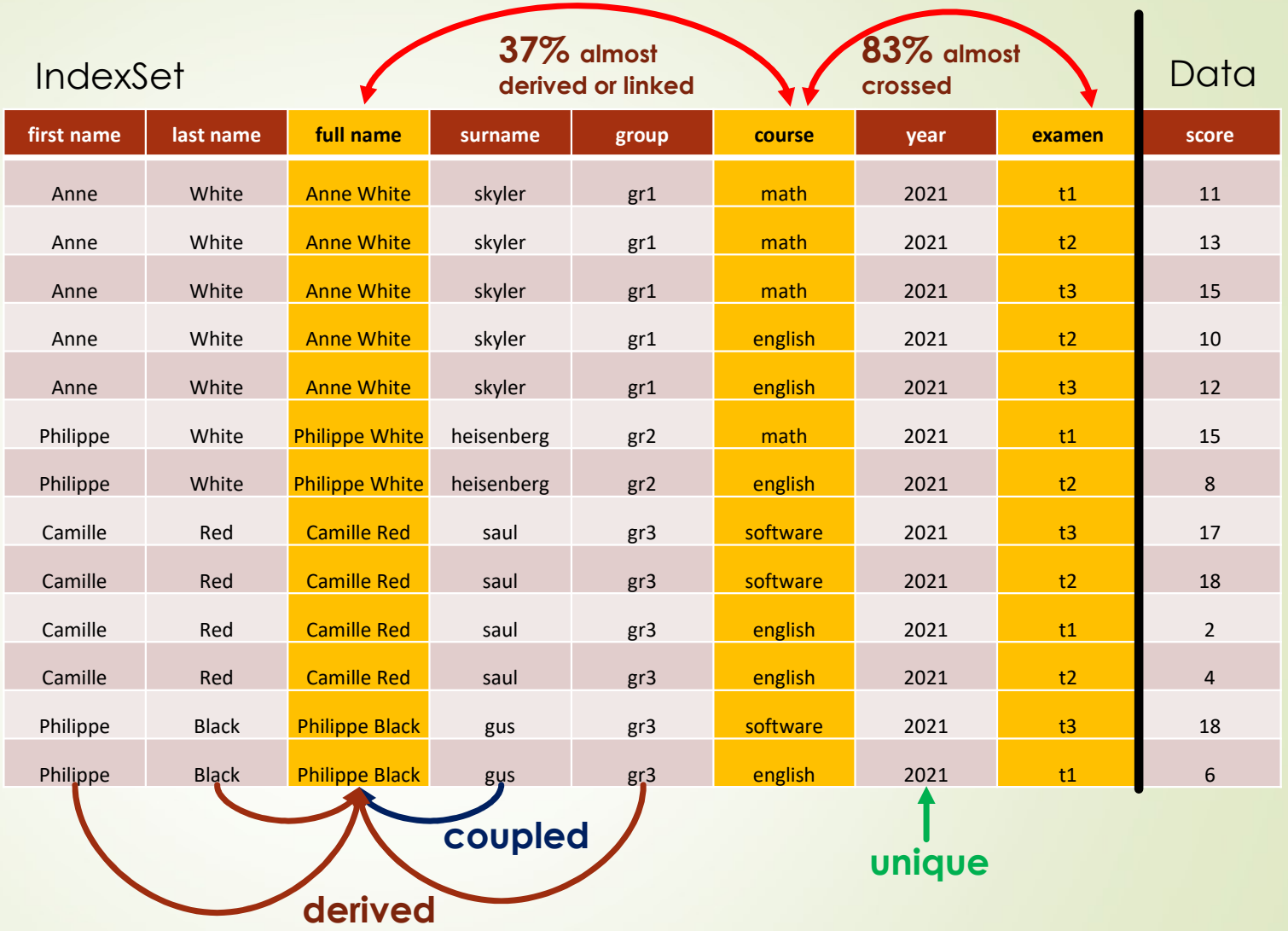
3 columns are linked
- Full name
- Course
- Examen

3 columns are derived
- First name
- Last name
- Group

1 column is coupled
- Surname

1 column is unique
- Year

ratio
- Name – Course : 37,5 %
- Name – Examen : 62,5 %
- Course – Examen : 83,7 %



first name	last name	full name	surname	group	course	year	examen	score
Anne	White	Anne White	skyler	gr1	math	2021	t1	11
Anne	White	Anne White	skyler	gr1	math	2021	t2	13
Anne	White	Anne White	skyler	gr1	math	2021	t3	15
Anne	White	Anne White	skyler	gr1	english	2021	t2	10
Anne	White	Anne White	skyler	gr1	english	2021	t3	12
Philippe	White	Philippe White	heisenberg	gr2	math	2021	t1	15
Philippe	White	Philippe White	heisenberg	gr2	english	2021	t2	8
Camille	Red	Camille Red	saul	gr3	software	2021	t3	17
Camille	Red	Camille Red	saul	gr3	software	2021	t2	18
Camille	Red	Camille Red	saul	gr3	english	2021	t1	2
Camille	Red	Camille Red	saul	gr3	english	2021	t2	4
Philippe	Black	Philippe Black	gus	gr3	software	2021	t3	18
Philippe	Black	Philippe Black	gus	gr3	english	2021	t1	6

1 - Functions

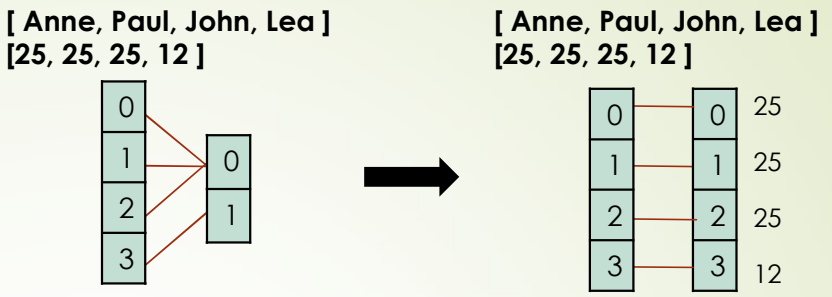
- **Derived to coupled**
  - Duplication of index key

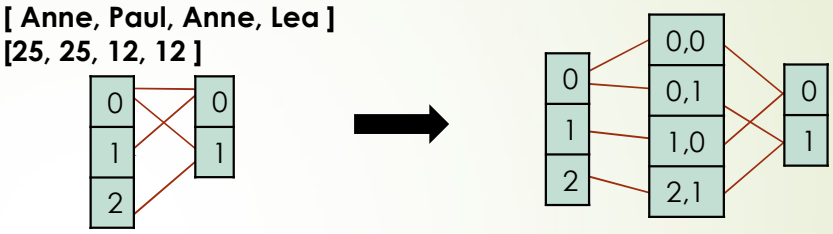
- **Index merging**
  - Index A and B are derived from Index (A,B)
    -> eg replace two primary indexes by one

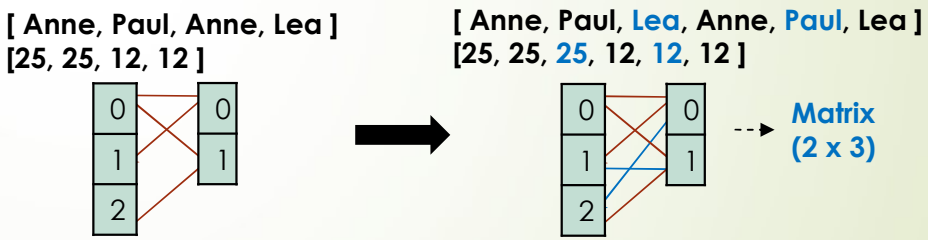
- **Linked to crossed**
  - Add link

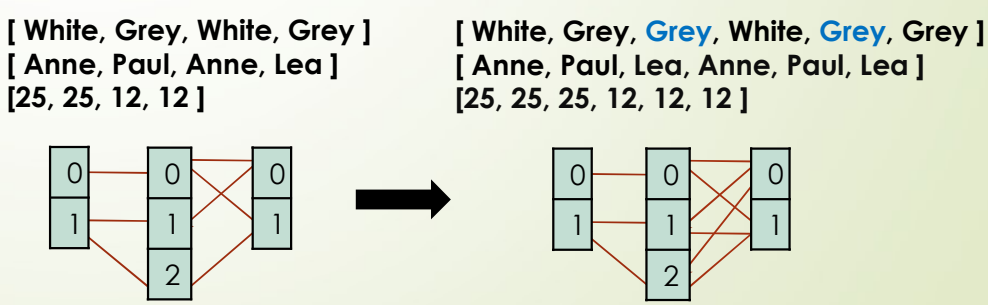
  (Link number = distmax)

- **Derived (coupled) extension**
  - Link propagation



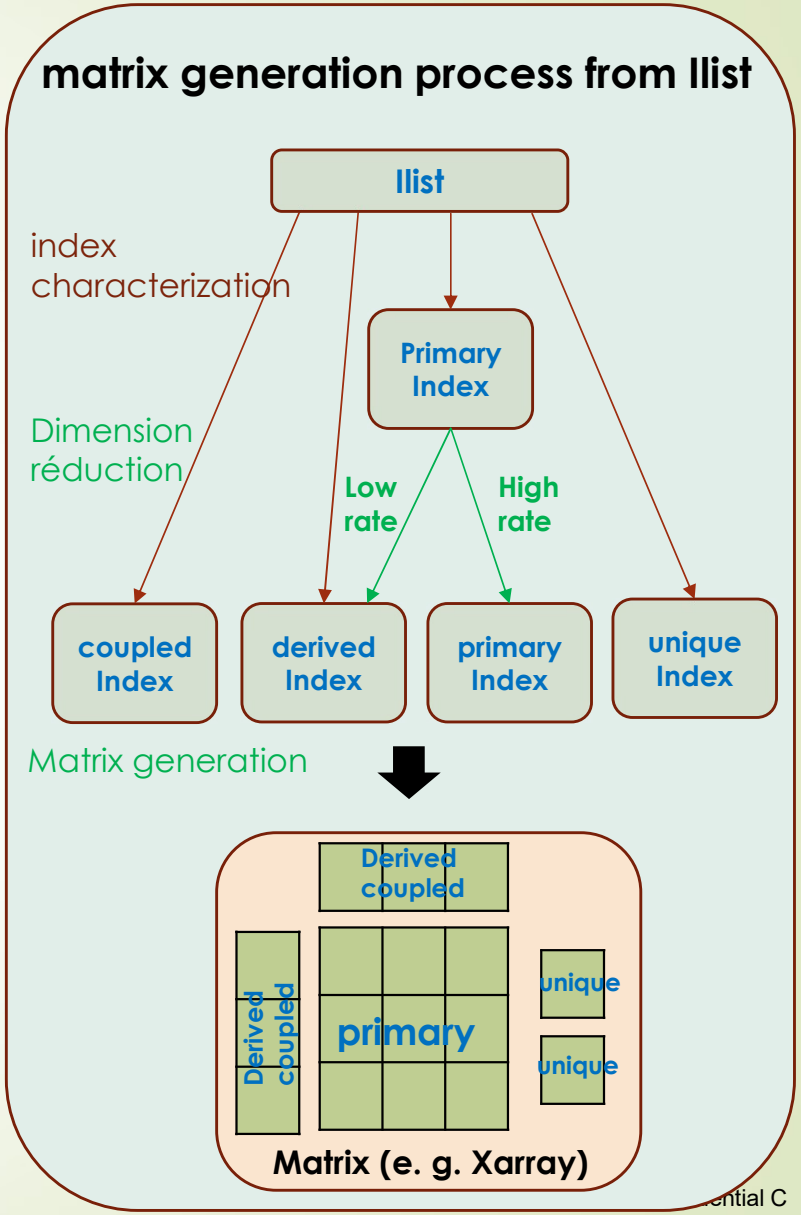






2 - Matrix generation process

- **Index characterization**
  - Identification of primary indexes
  - Association of coupled and derived indexes to primary indexes
- **Dimension reduction (if necessary)**
  - Primary index merging (rather low rate)
- **Matrix generation**
  - Full indexes conversion
    - Linked to crossed (primary indexes)
    - Extension (derived and coupled indexes)
  - Conversion
    - E.g. Xarray
      - Primary indexes -> dims
      - Derived/coupled indexes -> coords
      - Indexed value -> values
      - Unique index -> attrs



2 - Example

Full function :

- Axes are completed

Dimension :

- Canonical : 3
- Reduction : 2 (merge course, full name)

first name	last name	full name	surname	group	course	year	examen	score
Anne	White	Anne White	skyler	gr1	english	2021	t1	-
Anne	White	Anne White	skyler	gr1	english	2021	t2	10
Anne	White	Anne White	skyler	gr1	english	2021	t3	12
Anne	White	Anne White	skyler	gr1	math	2021	t1	11
Anne	White	Anne White	skyler	gr1	math	2021	t2	13
Anne	White	Anne White	skyler	gr1	math	2021	t3	15
Anne	***White***	***Anne White***	***skyler***	***gr1***	***software***	***2021***	***t1***	-
Anne	***White***	***Anne White***	***skyler***	***gr1***	***software***	***2021***	***t2***	-
Anne	***White***	***Anne White***	***skyler***	***gr1***	***software***	***2021***	***t3***	-

completed

derived

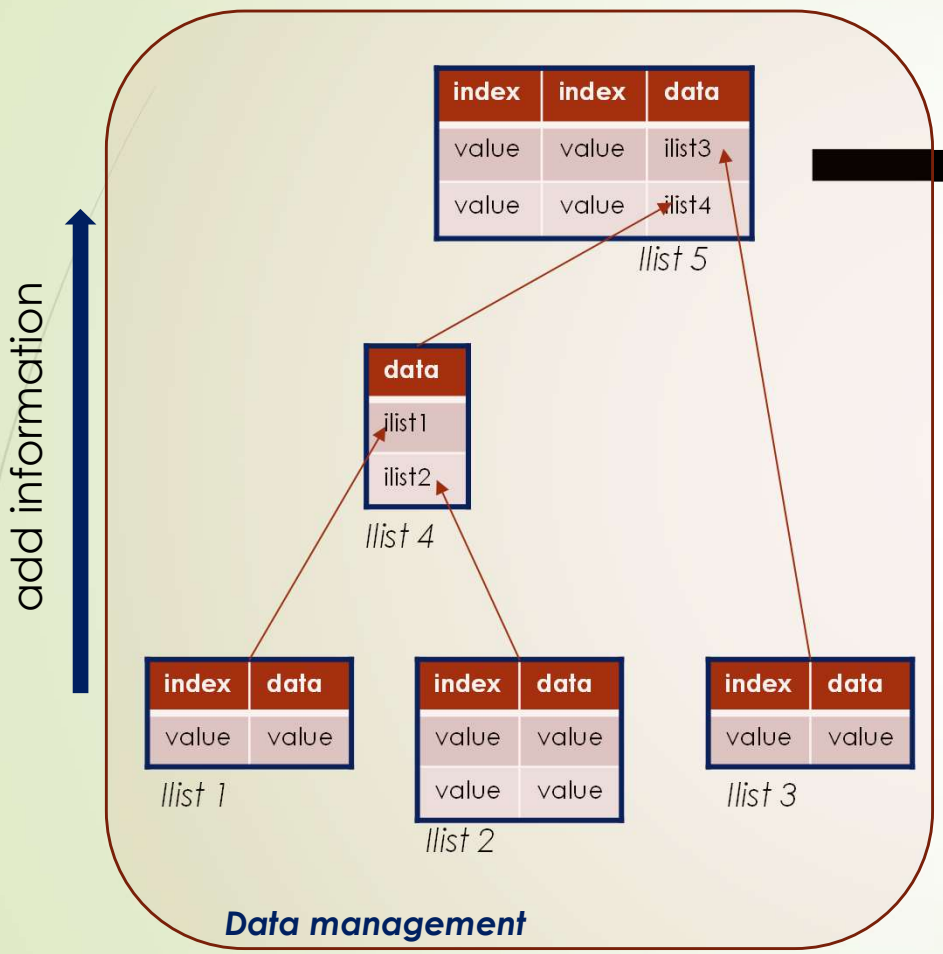
coupled

unique

```
In [366]: il.to_xarray(fillvalue=math.nan)
Out[366]:
<xarray.DataArray 'score' (full name: 4, course: 3, examen: 3)>
array([[[nan, 10., 12.],
        [11., 13., 15.],
        [nan, nan, nan]],

       [[ 2.,  4., nan],
        [nan, nan, nan],
        [nan, 18., 17.]],

       [[ 6., nan, nan],
        [nan, nan, nan],
        [nan, nan, 18.]],

       [[nan,  8., nan],
        [15., nan, nan],
        [nan, nan, nan]]])
Coordinates:
    first name  (full name) <U8 'Anne' 'Camille' 'Philippe' 'Philippe'
    last name   (full name) <U5 'White' 'Red' 'Black' 'White'
  * full name   (full name) <U14 'Anne White' 'Camille Red' ... 'Philippe White'
    surname     (full name) <U10 'skyler' 'saul' 'gus' 'heisenberg'
    group       (full name) <U3 'gr1' 'gr3' 'gr3' 'gr2'
  * course      (course) <U8 'english' 'math' 'software'
  * examen      (examen) <U2 't1' 't2' 't3'
Attributes:
    year:     2021
```

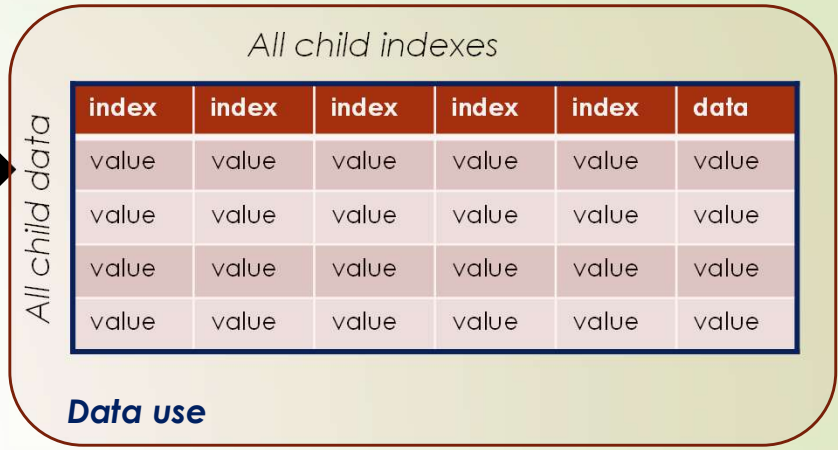
```
In [364]: il = Ilist.from_csv(filelight, delimiter=';', dtype=1)

In [365]: il.to_xarray(dimmax=2, fillvalue=math.nan)
Out[365]:
<xarray.DataArray 'score' (examen: 3, ["course", "full name"]: 8)>
array([[11., 15., nan, nan,  2.,  6., nan, nan],
       [13., nan, 10.,  8.,  4., nan, 18., nan],
       [15., nan, 12., nan, nan, nan, 17., 18.]])
Coordinates:
    first name                ([course, full name]) <U8 'Anne' ... 'Philippe'
    last name                 ([course, full name]) <U5 'White' ... 'Black'
    full name                 ([course, full name]) <U14 'Anne White' ... ...
    surname                   ([course, full name]) <U10 'skyler' ... 'gus'
    group                     ([course, full name]) <U3 'gr1' 'gr2' ... 'gr3'
    course                    ([course, full name]) <U8 'math' ... 'software'
  * examen                    (examen) <U2 't1' 't2' 't3'
  * ["course", "full name"]   ([course, full name]) <U6 '(0, 0)' ... '(2, 3)'
Attributes:
    year:     2021
```

3 - Aggregation process



Merge

(recursive function)



- **Process adapted to organizations**
- **Add information without altering**
- **Separation of management and use**

3 - Example

IndexSet | Data

aw

course	year	examen	score
math	2021	t1	11
math	2021	t2	13
math	2021	t3	15
english	2021	t2	10
english	2021	t3	12

pw

course	year	examen	score
math	2021	t1	15
english	2021	t2	8

cr

course	year	examen	score
software	2021	t3	17
software	2021	t2	18
english	2021	t1	2
english	2021	t2	4

pb

course	year	examen	score
software	2021	t3	18
english	2021	t1	6

total

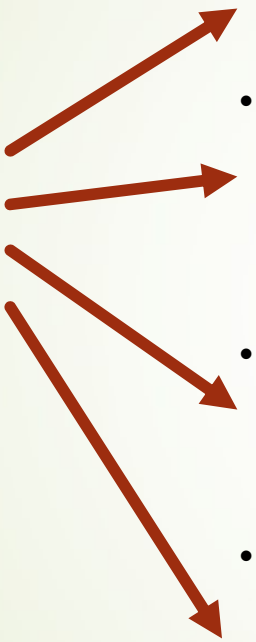
first name	last name	full name	surname	group	file
Anne	White	Anne White	skyler	gr1	aw
Philippe	White	Philippe White	heisenberg	gr2	pw
Camille	Red	Camille Red	saul	gr3	cr
Philippe	Black	Philippe Black	gus	gr3	pb

total.merge()

first name	last name	full name	surname	group	course	year	examen	score
Anne	White	Anne White	skyler	gr1	math	2021	t1	11
Anne	White	Anne White	skyler	gr1	math	2021	t2	13
Anne	White	Anne White	skyler	gr1	math	2021	t3	15
Anne	White	Anne White	skyler	gr1	english	2021	t2	10
Anne	White	Anne White	skyler	gr1	english	2021	t3	12
Philippe	White	Philippe White	heisenberg	gr2	math	2021	t1	15
Philippe	White	Philippe White	heisenberg	gr2	english	2021	t2	8
Camille	Red	Camille Red	saul	gr3	software	2021	t3	17
Camille	Red	Camille Red	saul	gr3	software	2021	t2	18
Camille	Red	Camille Red	saul	gr3	english	2021	t1	2
Camille	Red	Camille Red	saul	gr3	english	2021	t2	4
Philippe	Black	Philippe Black	gus	gr3	software	2021	t3	18
Philippe	Black	Philippe Black	gus	gr3	english	2021	t1	6

4 – format

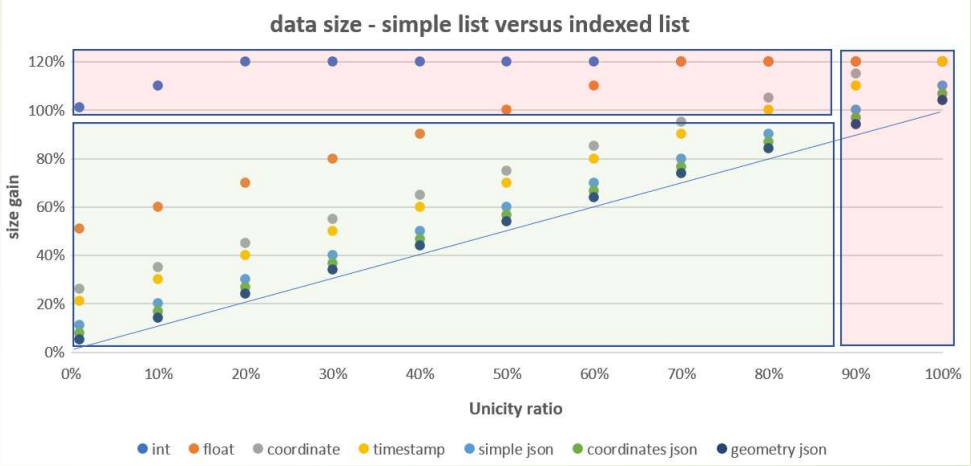
- **Ilist format**
  - Dict + Array



- **Tabular format (csv)**
  - Easy to read, duplication data, text only
- **Json format**
  - Easy to read, text only
  - Not duplication data
  - Compatible with NoSQL Database
- **Bson format**
  - Compatible with json format
  - Binary, structured data (eg datetime)
- **Binary format**
  - CBOR (Concise Binary Object Representation)
  - Compatible with json format
  - Binary, numerical, text, structured (eg datetime, coordinates)

4 – list size

- **Simple list size = n * l**
  - n : number of values
  - l : mean value size
- **Indexed list size = n * i + nx * l**
  - i : integer size
  - nx : number of different values
- **Indexed list size / list size = i / l (object lightness) + nx / n (unicity level)**
- **Properties**
  - If object lightness and unicity level are low, the indexed list size is lower than simple list size
    - e.g. : i / l = 0.1 , nx / n = 0.4 => indexed list size = 0.5 * list size
- **In a llist with data more complex than numerical data, the json (or binary) format has a smaller size than a tabular format**



Object lightness	l	i / l
int	2	1,00
float, int32	4	0,50
coordinate	8	0,25
string(10) (eg. timestamp)	10	0,20
simple json element (eg key/value)	20	0,10
structured json element (eg coordinates)	30	0,07
complex json element (eg geometry)	50	0,04

E.g. previous example :

- **csv : 2 418 bytes**
- **json : 1 496 bytes**
- **binary (CBOR) : 697 bytes**