Environnemental Sensing

Ilist

Concepts and principles

0 - Principles

1 - Index analysis

2 - Matrix generation

3 - Aggregation

4 – Format, storage

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

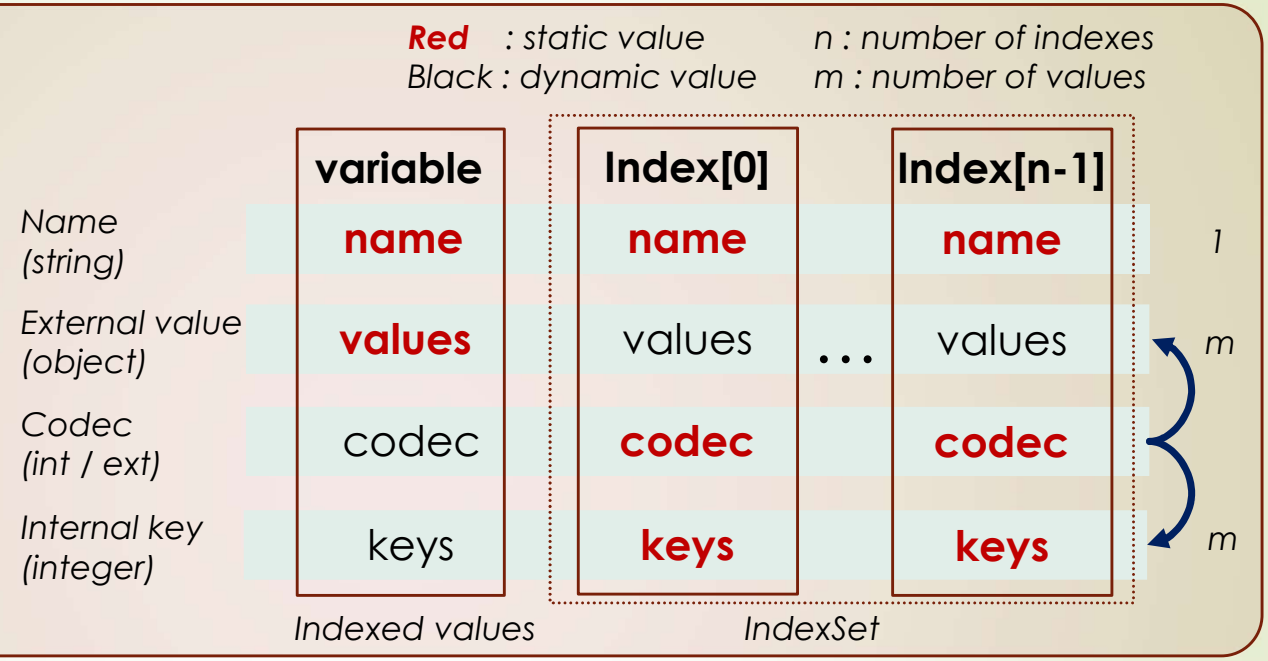
Example : csv file, measurement, log, matrix

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

codec

Internal key

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

name *values* *keys*

1 - Index categories

Values	v	**[Anne, Paul, Anne]**	**[Anne, Anne, Anne]**	**[Anne, Paul, Anne]**
Codec (row)	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

Max = len(values)
min = 1
x = len(codec)

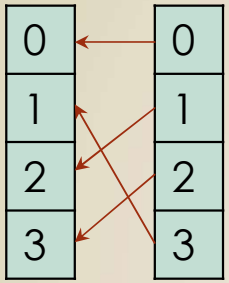
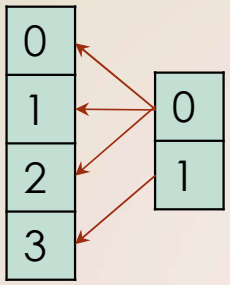
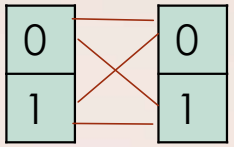
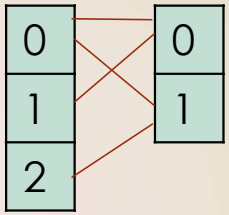
Rate : (M – x) / (M – m)
Dist to min : x - m
Dist to max : M - x

- **Definition**
  - Default codec : list of differents values
  - Full codec : list of values
- **Properties**
  - An index with full codec is complete
  - Any index have a default codec and a full codec
  - Default codec is the shortest, full codec is the longest

A codec defines the correspondence between values and keys (e.g.) :
- *1 : Anne*
- *0 : Paul*
- *2 : John*

A codec may not be bijective (e.g.) :
- *0 : Anne*
- *1 : Paul*
- *2 : Anne*

1 - linking categories

Values	v1 v2	[Anne, Paul, John, Lea] [25, 26, 15, 35]	[Anne, Paul, John, Lea] [25, 25, 25, 12]	[Anne, Paul, Anne, Paul] [25, 25, 12, 12]	[Anne, Paul, Anne, Lea] [25, 25, 12, 12]
Codec (row) i1 i2		 I2 coupled to i1	 I2 derived from i1		
Type		coupled (asymmetrical)	derived (asymmetrical)	crossed	linked
Property		Rate : 0 Disttomin : 0 diff : 0	Rate : 0 Disttomin : 0 0 < diff < min	Rate : 1 Disttomax : 0	0 < Rate < 1 min < dist < Max

Max = len(i1) * len(i2)
min = max(len(i1), len(i2)
diff = abs(len(i1) – len(i2))
x = len(index(v1, v2))

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

- **Properties**
  - Indicators are independant of values (length or value)
  - 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 to B, all the relationships with other indexes are identical

1 - Example

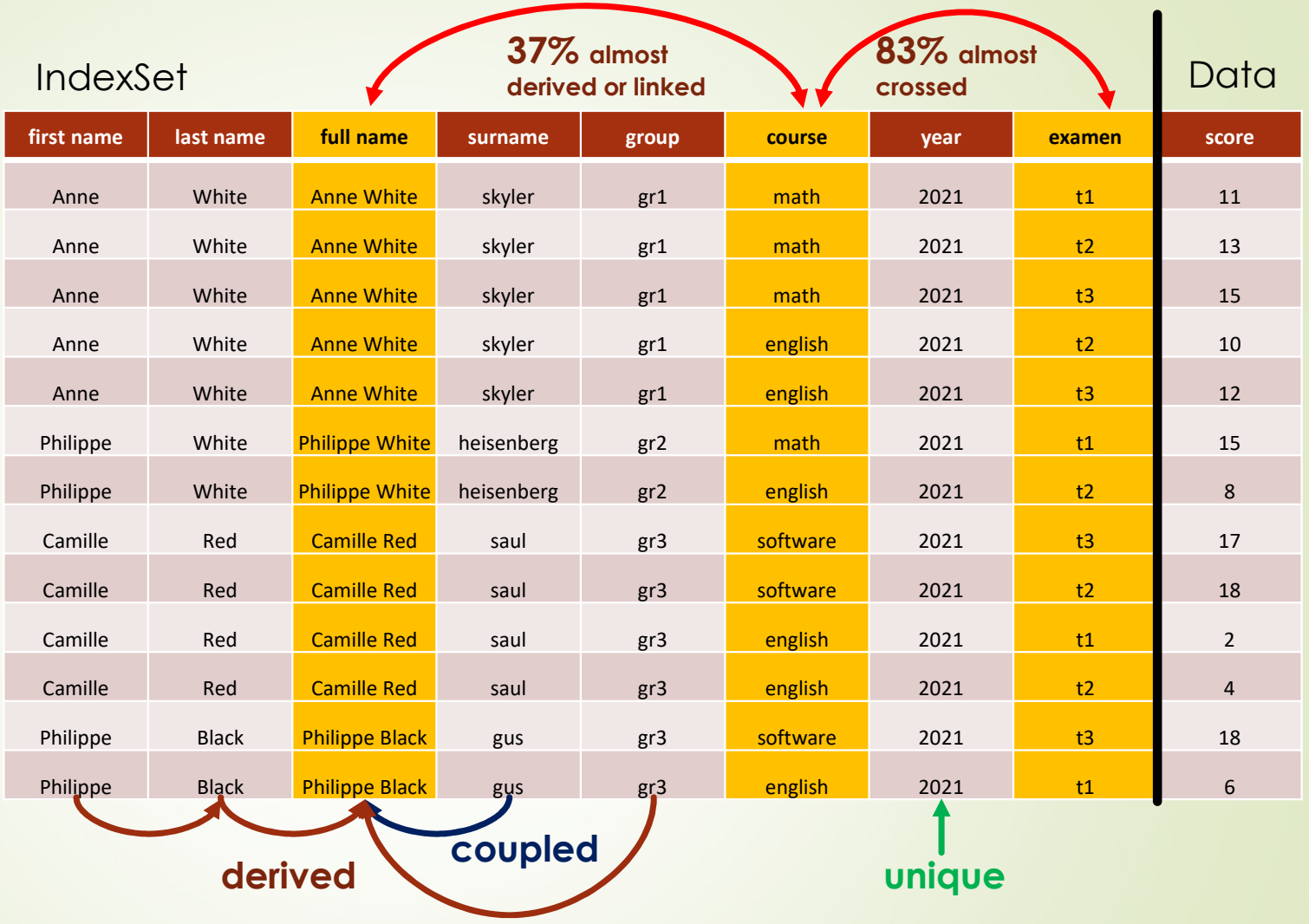
3 columns are linked
- Full name
- Course
- Examen

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

1 column is coupled
- Surname

1 column is unique
- Year

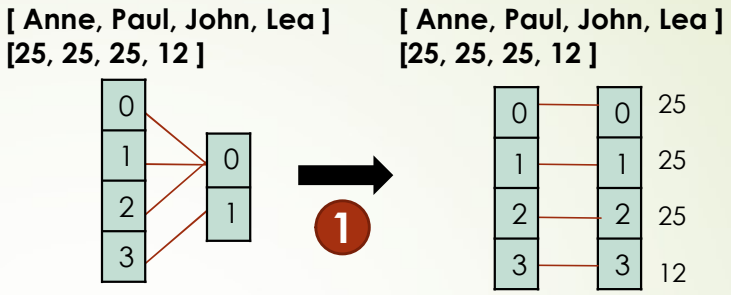
ratio
- Name – Course : 37,5 %
- Name – Examen : 62,5 %
- Course – Examen : 83,7 %

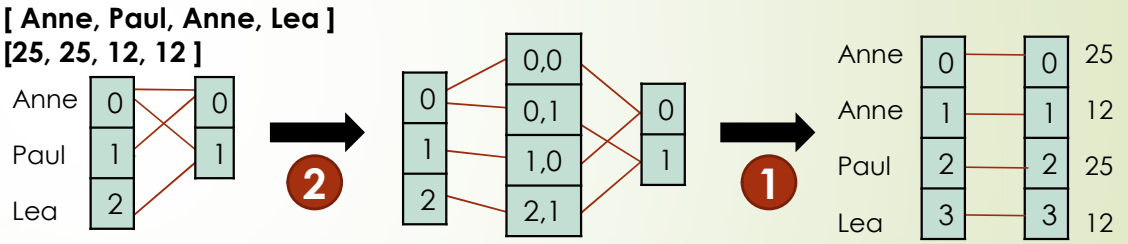


first name	last name	full name	surname	group	course	year	examen	score
Anne	White	Anne White	skyler	gr1	math	2021	t1	11
Anne	White	Anne White	skyler	gr1	math	2021	t2	13
Anne	White	Anne White	skyler	gr1	math	2021	t3	15
Anne	White	Anne White	skyler	gr1	english	2021	t2	10
Anne	White	Anne White	skyler	gr1	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 – Codec extension

- **Derived to coupled**
  - Extension of index codec
- **Derived to derived**
  - Minimale extension with all keys information (see next slide)
- **Coupling (linked to coupled)**
  - Index A and B are derived from Index (A,B)
    -> eg replace two primary indexes by one

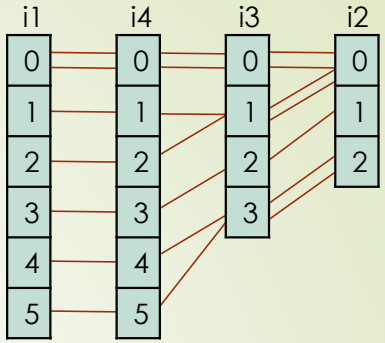




- **Properties**
  - All indexes can be transformed into coupled index (codec extension)
  - csv data is all composed of coupled indexes
  - A derived index is defined only by his extended codec (no duplication)
  - An index derived from multiple indexes has minimal codec if the dist_to_max is minimal

1 – Derived to derived

- **Method**
  - Keys can be generated with codec and reference to parent index
  - First derived index is convert in coupled index
  - Codec is minimal



- I1 : [d1, d1, d2, d3, d4, d5, d6]
  [0, 0, 1, 2, 3, 4, 5]
  [d1, d2, d3, d4, d5, d6]
- I4 : [j, j, f, f, m, s, s]
  [0, 0, 1, 1, 2, 3, 3]
  [j, f, m, s]
- I3 : [t1, t1, t1, t1, t2, t3, t3]
  [0, 0, 0, 0, 1, 2, 2]
  [t1, t2, t3]
- I2 : [s1, s1, s1, s1, s1, s2, s2]
  [0, 0, 0, 0, 0, 1, 1]
  [s1, s2]

Derived E = 2

Derived E = 1

Derived E = 1

Coupled E = 0

Derived E = 2

Derived E = 1

[d1, d1, d2, d3, d4, d5, d6]
[0, 0, 1, 2, 3, 4, 5]
[d1, d2, d3, d4, d5, d6]

[j, j, f, f, m, s, s]
[0, 0, 1, 2, 3, 4, 5] 3
[j, f, f, m, s, s]

[t1, t1, t1, t1, t2, t3, t3]
[0, 0, 1, 1, 2, 3, 3] 2
[t1, t1, t2, t3]

[s1, s1, s1, s1, s1, s2, s2]
[0, 0, 0, 0, 1, 2, 2] 1
[s1, s1, s2]

[d1, d1, d2, d3, d4, d5, d6]
[0, 0, 1, 2, 3, 4, 5]
[d1, d2, d3, d4, d5, d6]

[j, j, f, f, m, s, s]
[0, 0, 1, 2, 3, 4, 5] 1
[j, f, f, m, s, s]

[t1, t1, t1, t1, t2, t3, t3]
[0, 0, 1, 1, 2, 3, 3] 2
[t1, t1, t2, t3]

[s1, s1, s1, s1, s1, s2, s2]
[0, 0, 0, 0, 1, 2, 2] 3
[s1, s1, s2]

Default codec

Calculation of the extended codec

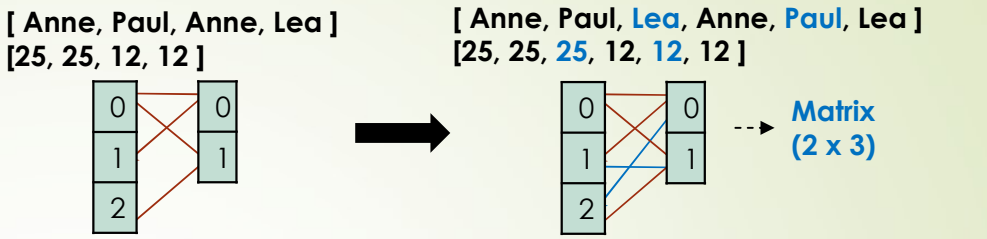
Calculation of keys

Reset index

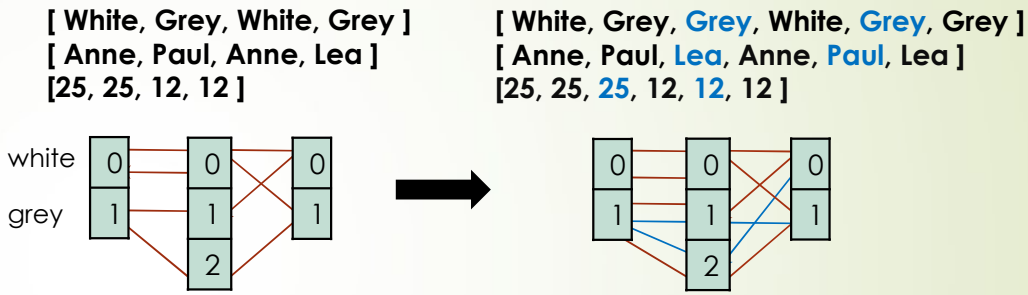
1 – Variable extension

- **Linked (derived, coupled) to crossed**
  - Add link (Variable extension)

  (Link number = dist to max)



- **Derived (coupled) extension**
  - Link propagation is obtain with « derived to coupled » function

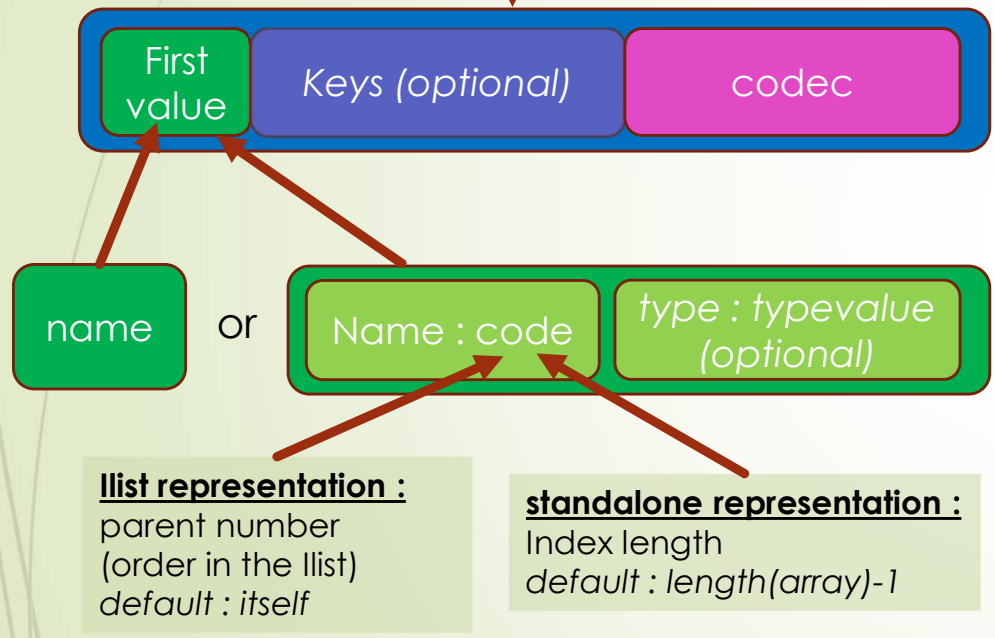


- **Properties**
  - Link propagation is available for all derived or coupled indexes
  - All indexes can be transformed into crossed index (add values in Variable)
  - Extension is impossible with linked index
  - matrix data is equivalent to crossed indexes

1 – Representation

If length(array) <= len(index) + 1
only codec
If length(array) > len(index) + 1
keys + codec

JSON Array



Ilist representation :
parent number
(order in the Ilist)
default : itself

standalone representation :
Index length
default : length(array)-1

Format
Text (json), Binary (CBOR)

Example : 'name'

['Anne', 'Anne', 'John', 'Paul', 'John']

- **Full format (standalone)**
  ['name', 'Anne', 'Anne' 'John', 'Paul', 'John']
  -> Full codec (e.g. csv format)
- **Default format (standalone)**
  [{'name': 5}, 0,0,1,2,1,'Anne', 'John', 'Paul']
  -> Default codec, keys
- **Default format (Ilist - not complete)**
  ['name', 0,0,1,2,1,'Anne', 'John', 'Paul']
  -> Default codec, keys
- **Default format (Ilist - primary)**
  ['name', 'Anne', 'John', 'Paul']
  -> Default codec, variable extension
- **Extended format (Ilist – derived or coupled)**
  [{'name': 2}, 'Anne', 'John', 'Paul', 'John']
  -> Extended codec, derived index

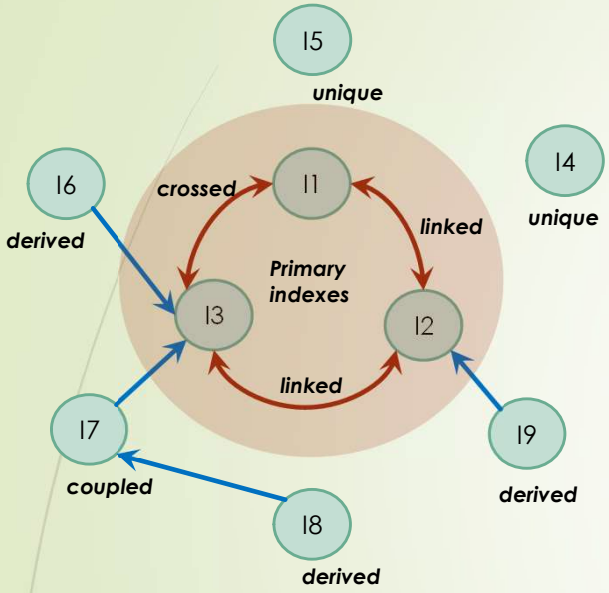
2 – IndexSet (list of indexes)

- **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
  - The parent index is the index with the lowest diff number in the list of coupling or derivating indexes (or itself if the index is primary)
  - The precursor index is the Primary index in the indexing tree

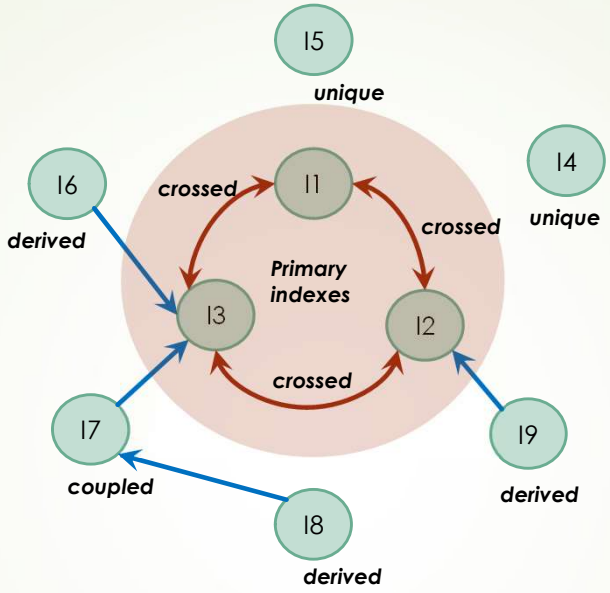
- **IndexSet definition**
  - Dimension : number of primary indexes
  - Complete : An indexSet is complete if all the primary indexes are crossed with each other primary index

- **Properties**
  - **The number of values of a full indexset is the product of the primary indexes length**
  - **A complete IndexSet can be transformed in a Matrix with the dimension of the indexset**
  - **Keys data is unnecessary in a complete indexset if derived codec are extended**
  - **Dimension can be reduced by index extension**
  - **Dimension can be increased by variable extension**

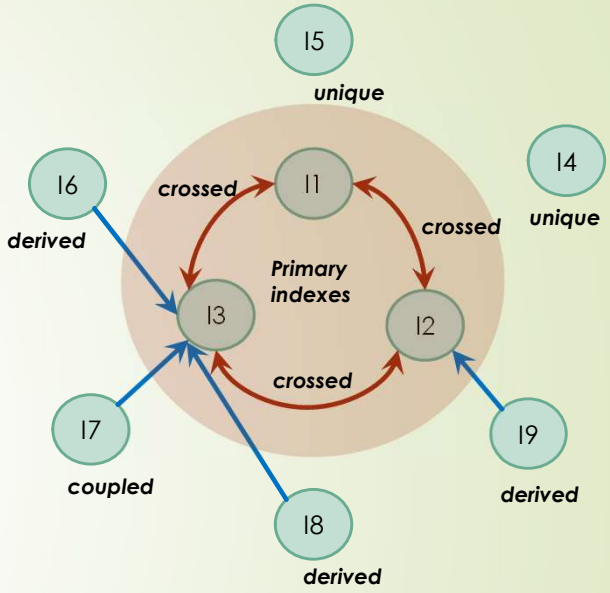
2 – Format



Canonical format



Complete (parent indexes)

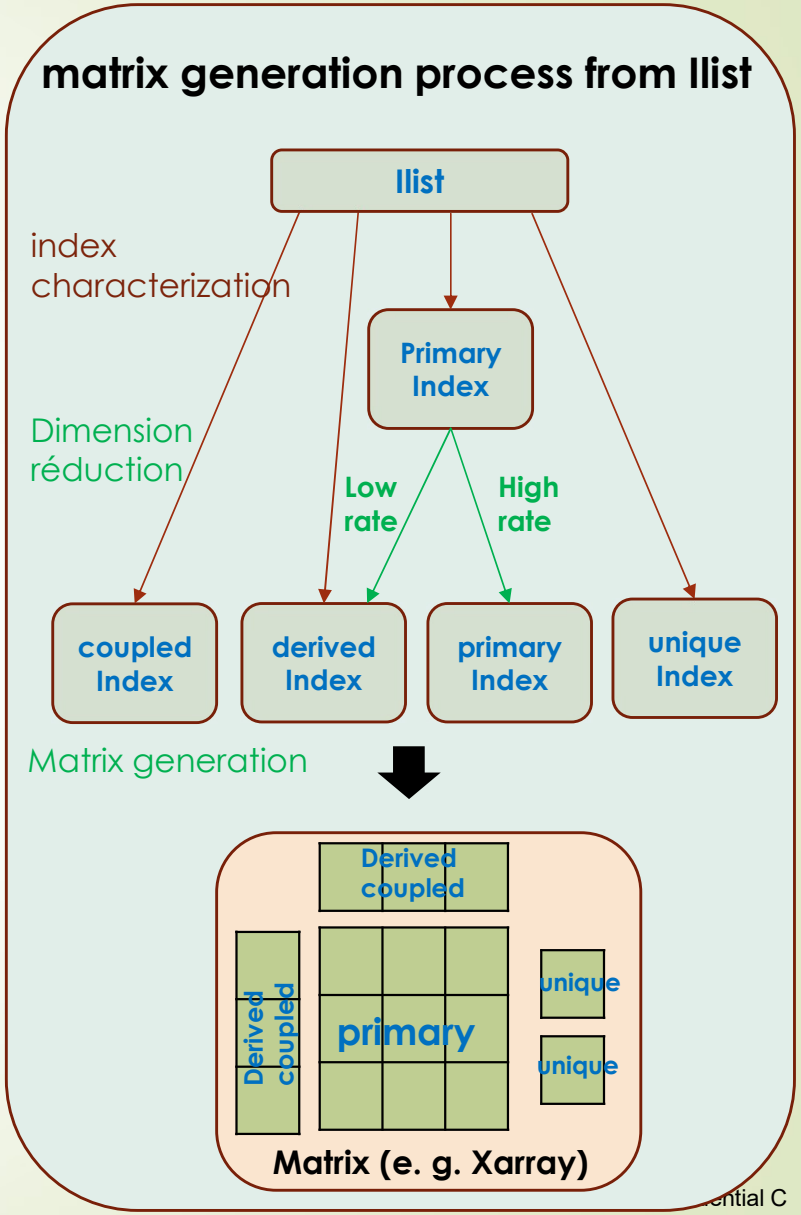


Complete (precursor indexes)

- **Properties**
  - Each indexset has a canonical format (at least one primary index)
  - Complete data is obtained by crossing all the primary indexes (variable extension)
  - Exchange format can contains only extended codec data
  - Complete indexset can be transformed in Matrix
  - Csv format is a canonical format with one primary index and any coupled indexes, all indexes have full codec

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 -> data
      - Unique index -> attrs



2 - Example

Full function :

- Primary are completed
- Derived are full codec

completed

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***	-

derived

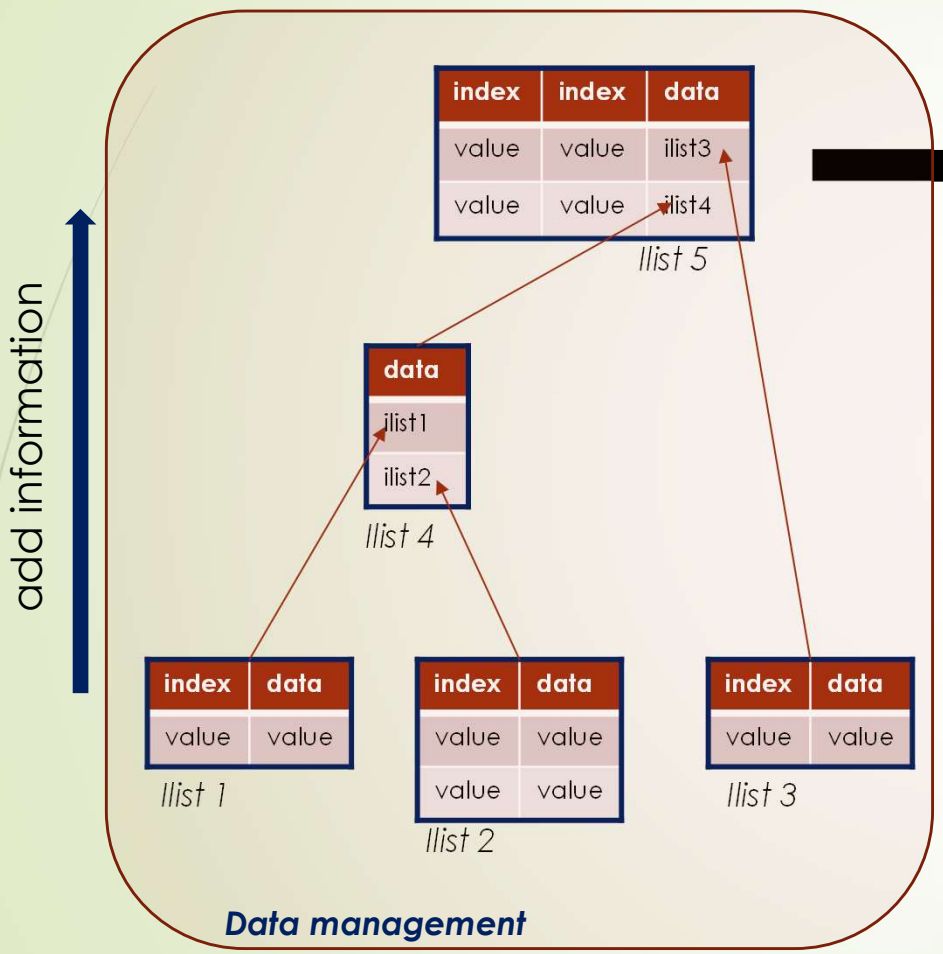
coupled

unique

```
In [367]: cours.to_xarray(axes=cours.axesmin)
Out[367]:
<xarray.DataArray 'Ilist' (full name: 4, course: 3, examen: 3)>
array([[['?', '10', '12'],
        ['11', '13', '15'],
        ['?', '?', '?']],

       [['2', '4', '?'],
        ['?', '?', '?'],
        ['?', '18', '17']],

       [['6', '?', '?'],
        ['?', '?', '?'],
        ['?', '?', '18']],

       [['?', '8', '?'],
        ['15', '?', '?'],
        ['?', '?', '?']]], dtype='<U2')
```

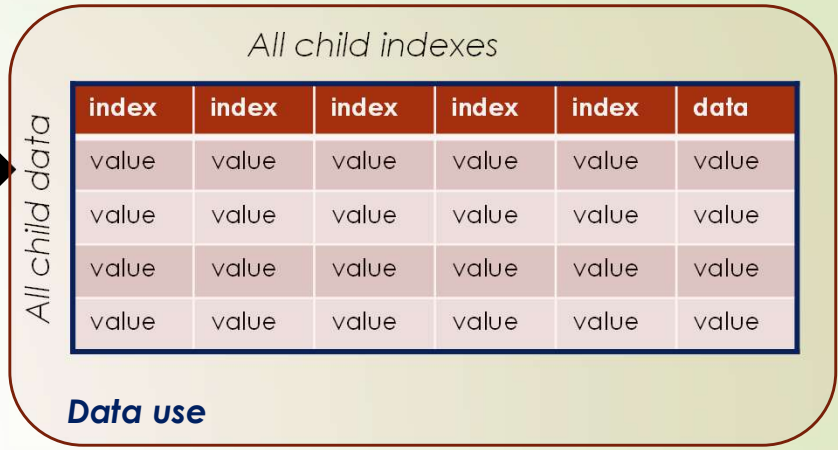
```
Coordinates:
    ï»¿first name  (full name) <U8 'Anne' 'Camille' 'Philippe' 'Philippe'
    last name      (full name) <U5 'White' 'Red' 'Black' 'White'
  * full name      (full name) <U14 'Anne White' ... 'Philippe White'
    surname        (full name) <U10 'gus' 'heisenberg' 'saul' 'skyler'
    group          (full name) <U3 'gr1' 'gr3' 'gr3' 'gr2'
  * course         (course) <U8 'english' 'math' 'software'
  * examen         (examen) <U2 't1' 't2' 't3'
```

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

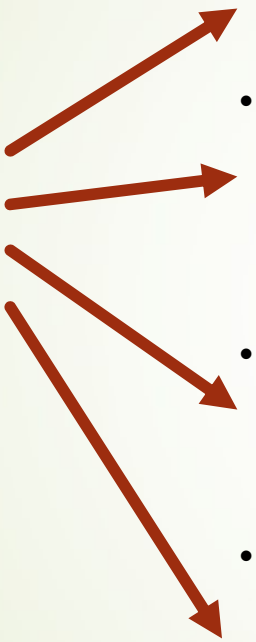
first name	last name	full name	surname	group	file
Anne	White	Anne White	skyler	gr1	aw
Philippe	White	Philippe White	heisenberg	gr2	pw
Camille	Red	Camille Red	saul	gr3	cr
Philippe	Black	Philippe Black	gus	gr3	pb

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

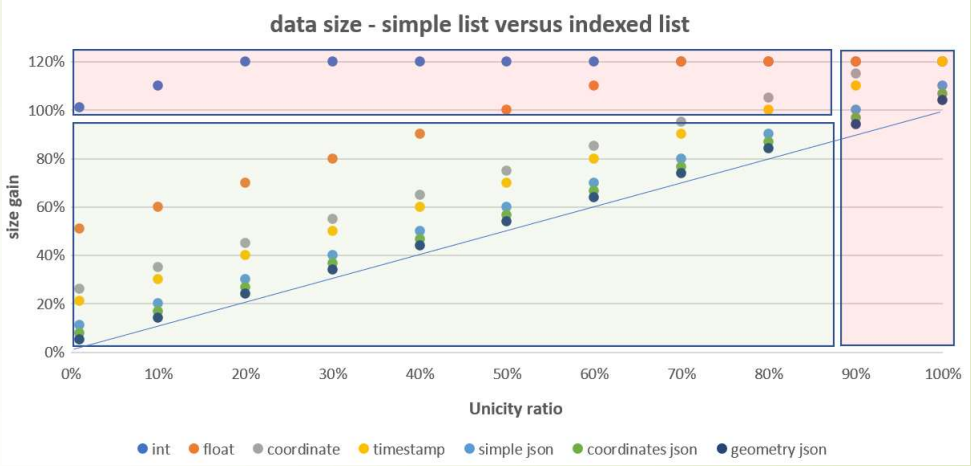
- **Ilist format**
  - Dict + Array



- **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**

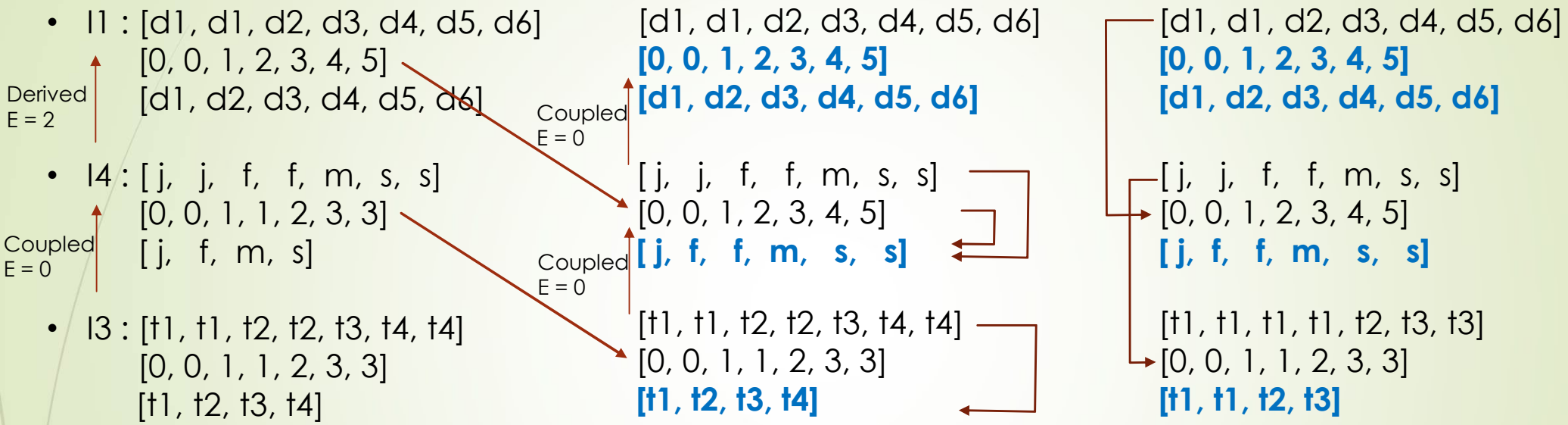


Object lightness	l	i / l
int	2	1,00
float, int32	4	0,50
coordinate	8	0,25
string(10) (eg. timestamp)	10	0,20
simple json element (eg key/value)	20	0,10
structured json element (eg coordinates)	30	0,07
complex json element (eg geometry)	50	0,04

E.g. previous example :

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

1 – Derived indexes



Default codec

Calculation of the extended codec

Calculation of keys