Example of ZigZag thinking:

Classifying my book collection and deciding what to donate to the Internet Archive for scanning into OpenLibrary
OpenLibrary has an API that can tell you if they have an ISBN number, which can be scanned from a barcode.
Some books do not have ISBNs, so they require an additional search to see if Open Library has it.

Cells: an append-only log, contents are immutable
1 = Person, Eric
2 = Organization, OpenLibrary
3 = Organization, Local library
4 = Book, ISBN 101
5 = Book, ISBN 102
6 = Book, ISBN 103
7 = Person, Ted Nelson
8 = Person, Herman Melville
9 = Book, ISBN 201
10 = Book, ISBN 202
11 = Book Title: Possiplex
12 = Book Title: Theory of Hypertext
13 = Book Title: Geeks Bearing Gifts
14 = Book Title: Moby Dick
15 = Book Title: Bartleby
16 = Book Title: Punch Card Manual
17 = Book Title: History textbook from 1910
18 = Type: Person
19 = Type: Organization
20 = Type: Book
21 = Book Title: The missing manuscript
22 = Type: ISBN
23 = Book, ISBN 999 (A book in OpenLibrary that I do not have)
24 = Book, History textbook from 1801 (A book in OpenLibrary without an ISBN that I do not have)

Dimensions are also append-only logs. Pointers track sequence of cells. Version tracking can be done - not shown here.
Zigzag does not impose any structure or constraints - it is based on user-defined conventions.

d.Person
18, 1, 7, 8

d.Organization
19,2,3

d.ISBN
22,4, 5, 6, 9, 10, 23

d.Book (convention: If it has ISBN, use that, else use Book Title. Use of composite key of Title and Author not addressed here)
20,4,5,6,9,10,16,17,21,23,24

d.BookTitle
4,11, 5,12,   6,13,   9,14,   10,15,   16,   17,   21

d.BookAuthor   (books with multiple authors not in scope for my current problem domain, but can be modeled in ZigZag)
7, 4, 5, 6, 21,   8, 14, 15

[alternatively, the following dimension may be defined, or implemented as a view/ function - it is less normalized
d.BookInfo (convention: ISBN (if available), Author (if available), Book Title)
4,7,11, 5,7,12   6,7,13. 9,8,14. 10,8,15   16,   17,   7,21,   23,   24
]

d.ISBN can be fed into an API to determine which of my books with ISBNs are in OpenLibrary

d.BookPossession: (convention: Person, Book list)
1,4,5,6,9,10,16,17,21   2,4,6,9,16,23,24

Views or Functions: (functions can return a new list of cells)

Books I have that OpenLibrary has
1,2,4,6,9,16
        Do I still want it? Yes, keep it.
1,4,16
        NO: Donate to local library
3,6,9

Books I have that OpenLibrary doesn't have
1,5,10,17,21
        Do I still want it? Yes keep it.
5,21
        NO: Donate to OpenLibrary
10,17

Books OpenLibrary has that I don't have
23, 24, and millions more.