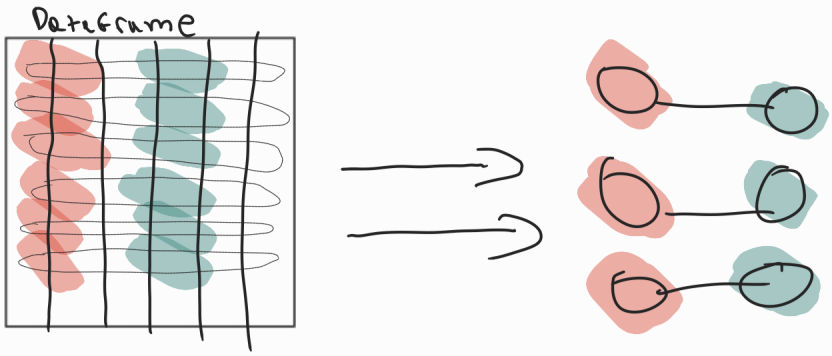
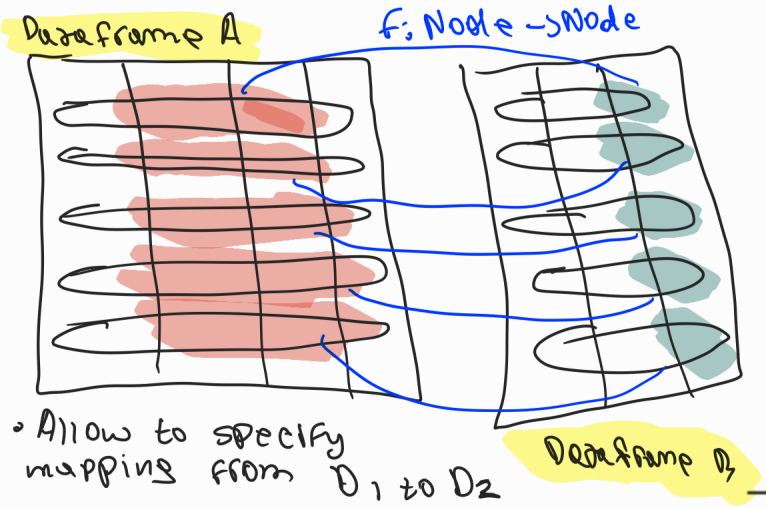
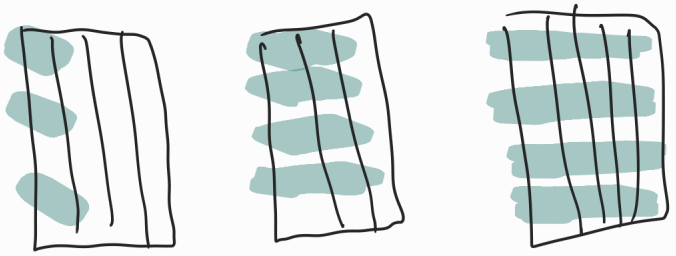




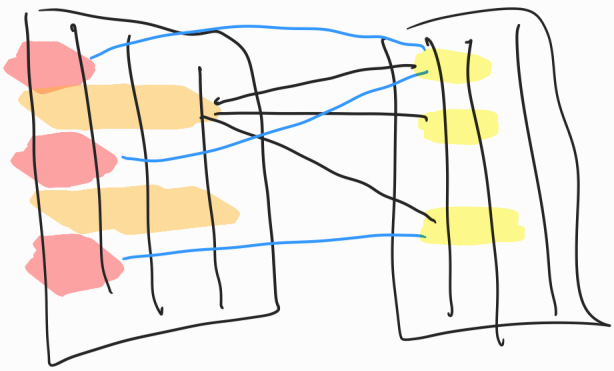
- Allow to specify mapping from D_1 to D_2

Node Selection



- Select which nodes to include by providing a predicate.
- Specifying how many attributes to include in a node.

Edge Selection



- Edge is a mapping from Domain of nodes to another Domain of nodes

High level execution

1 Form nodes

2 Form edges

3 Convert graph into cypher

Constant information:

- Nodes of different types
- Nodes form domain(s)
- Edges of different types
  - Edges can have other sort of metadata that reflect the procedure used to generate them
    - Different linking methods

Changing info.

None
- Once nodes or edges are formed no change on the ADT is done.

Data Definitions

- Node
- Edge
- Graph

Node:

→ It has a name or type. It allows it to be distinguished from other nodes.

→ It has attributes or properties that are derived or selected from a record.

→ It can be connected to other nodes with edges.

Edge:

- It has a name of type that tells about the rule used.
- It has meta that tells about the output of properties of a linking function used.
- It links two nodes.

Graph:

- It keeps track of Nodes and edges.
- It is an intermediate representation that can be used to_cypher, to_pandas.

1 Forming nodes

- We can have different Node types.

Node$_1$: [a_1, a_2 .... a_n]

Node$_2$: [j_1, j_2, ..., j_n]

form_nodes_from(df,

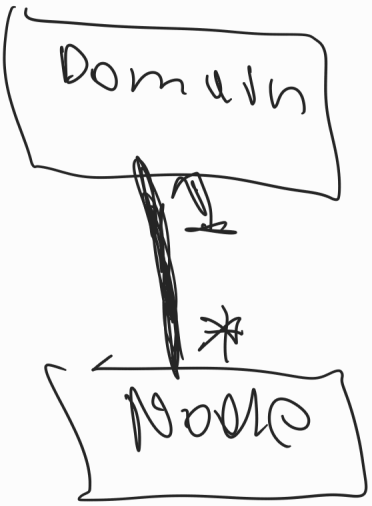
{Node$_1$: [...],

Node : [...],

...}) → [df_1, df_2]

List of dfs corresponding nodes.

It outputs different domains of nodes



2 Forming Edges

d_i : domain i

form_edge(d_1, d_2) →

Edges

→ How do we represent it?

↳ It depends on how we want to connect it to types.

n_i : node i

edge_func(n_1, n_2)

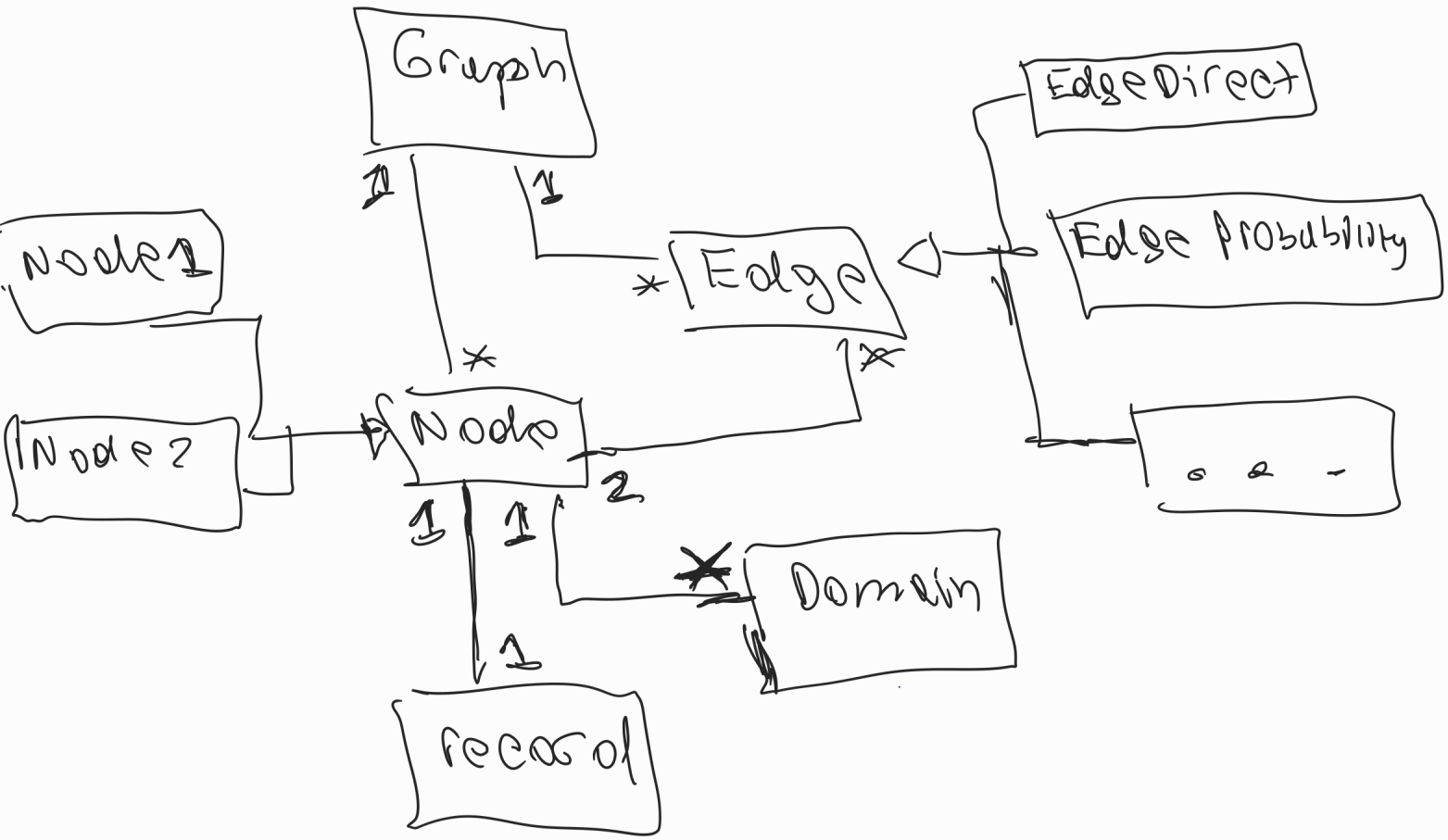
→ Edge

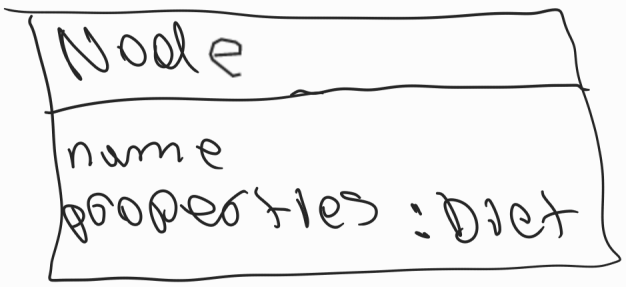
↑

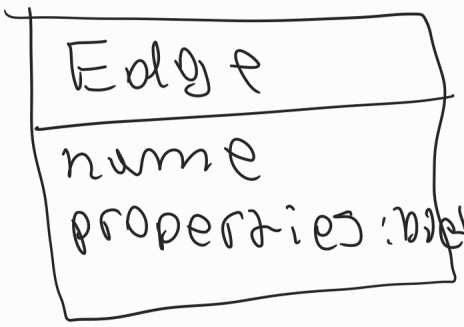
Edge datatype

3 Forming Graph

form_graph (nodes, edges) ---> Graph







4 Domain

```
form_domain(df, selection_criteria)
-> Domain
  selection_criteria(row) -> Node
```