

```
chain_of(
  with_elements(load_postgres_table(("public", "patient"), ["id"], [Int32])),
  flatten())
```



The diagram illustrates a complex computational graph, likely representing a neural network architecture or a data processing pipeline. The graph is composed of several interconnected nodes and edges, with a large blue shaded region and a large orange shaded region.

**Key Components and Operations:**

- load\_table("patient", ["id"]) SELECT id FROM patient:** This operation is shown in a green box within the blue shaded region. It is connected to a "head" node (yellow box) and a "1" node (yellow box).
- load\_table("patient", ["mn"], ["id"]) SELECT mn FROM patient WHERE id = ?:** This operation is shown in a green box within the blue shaded region. It is connected to a "head" node (yellow box) and a "1" node (yellow box).
- cardinality(x1to1):** This operation is shown in a green box within the orange shaded region. It is connected to a "head" node (yellow box) and a "1" node (yellow box).
- output():** This operation is shown in a green box. It is connected to a "head" node (yellow box) and a "1" node (yellow box).
- column(1):** This operation is shown in a green box. It is connected to a "head" node (yellow box) and a "1" node (yellow box).
- BlockOf, EntityShape, TupleOf, Int32, String:** These are the output types or shapes of the operations, shown in various colored boxes (blue, yellow, orange, green) and connected by dotted lines.

The graph shows a complex flow of data and operations, with many nodes and edges. The blue shaded region contains the main data processing operations, while the orange shaded region contains the output and cardinality operations. The right side of the image shows a detailed view of the operations, with nodes like BlockOf, EntityShape, TupleOf, and Int32, connected by dotted lines.







```
chain_of(with_elements(load_postgres_table(("public", "patient"), ["id"], [Int32])),
  flatten(),
  with_elements(
    chain_of(
      load_postgres_table(("public", "patient"), ["mrn"], [String], ["id"]),
      block_cardinality(x1to1))),
  flatten()),
  with_elements(
    chain_of(
      output(),
      column(1))))
```



















