

```
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 Flow:

- Inputs:** The graph starts with inputs from the left, including a "head" node and a "1" node, which feed into a "load_table" operation.
- Load Table Operations:** Two "load_table" operations are shown, both taking "patient" as input. The first "load_table" operation outputs to a "head" node, which then feeds into a "1" node. The second "load_table" operation outputs to a "head" node, which then feeds into a "1" node.
- Cardinality and Output:** A "cardinality(x1to1)" operation is shown, which feeds into a "head" node. This "head" node then feeds into a "1" node. The "1" node then feeds into an "output()" operation.
- Column and Final Output:** A "column(1)" operation is shown, which feeds into a "head" node. This "head" node then feeds into a "1" node. The "1" node then feeds into a "column(1)" operation, which finally outputs to a "String" node.
- Shaded Regions:** A large blue shaded region covers the central part of the graph, including the "load_table" operations and the "cardinality" operation. A large orange shaded region covers the right side of the graph, including the "output()" operation and the "column(1)" operation.
- Connections:** Dotted lines connect various nodes to a series of "BlockOf" and "EntityShape" nodes on the right, which then lead to "TupleOf" and "String" nodes.

The graph shows a complex flow of data and operations, with multiple paths and interactions between different components. The shaded regions highlight specific areas of interest within the overall architecture.







```
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))))
```















