

```
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**: A query operation that loads data from a table named "patient" and selects the "id" column.
- load_table("patient", ["mn"], ["id"]) SELECT mn FROM patient WHERE id = ?**: A query operation that loads data from a table named "patient" and selects the "mn" column, filtered by the "id" column.
- cardinality(x1to1)**: An operation that calculates the cardinality of the input.
- output()**: An operation that outputs the result of the computation.
- column(1)**: An operation that extracts the first column from the input.
- BlockOf**: A node representing a block of data, often followed by a dimensionality specification (e.g., **BlockOf x1to1**).
- EntityShape**: A node representing the shape of an entity, often followed by a dimensionality specification (e.g., **EntityShape "patient"**).
- TupleOf**: A node representing a tuple of data, often followed by a dimensionality specification (e.g., **TupleOf**).
- Int32**: A node representing an integer value.
- String**: A node representing a string value.

The graph shows a flow from inputs (e.g., **head**, **1**) through various operations and blocks, leading to a final output. The blue shaded region highlights a specific part of the graph, while the orange shaded region highlights another. The right side of the image provides a detailed view of the operations, showing the flow of data through various blocks and shapes.







```
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))))
```















