

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



```
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))))
```









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, organized into a hierarchical structure.

Key Components and Operations:

- Input/Output Nodes:** The graph starts with an input node (a small blue square) and ends with multiple output nodes, including `Int32` and `String` types.
- Core Operations:**
 - `load_table("patient", ["id"]) SELECT id FROM patient`: A query operation that loads data from a table.
 - `load_table("patient", ["mn"], ["id"]) SELECT mn FROM patient WHERE id = ?`: A query operation that loads data from a table, filtered by a specific ID.
 - `cardinality(x1to1)`: An operation that calculates the cardinality of a set.
 - `flatten()`: An operation that flattens a multi-dimensional array into a single dimension.
 - `output()`: An operation that outputs the result of the computation.
 - `column(1)`: An operation that extracts a specific column from a dataset.
- Tensor Shapes and Data Flow:**
 - `BlockOf x1to1`, `BlockOf x0toN`, `EntityShape "patient"`, and `TupleOf` are used to represent different data structures and their relationships.
 - The graph shows a flow from input data through various operations, resulting in intermediate shapes like `BlockOf x1to1` and `EntityShape "patient"`, which are then combined into `TupleOf` structures.
- Control Flow and Logic:**
 - The graph includes conditional logic (e.g., `IF` statements) and loops (e.g., `FOR` loops) that determine the flow of data and the execution of operations.
 - Nodes like `head` and `1` are used to represent specific data points or indices within the graph.

The overall structure suggests a complex data processing pipeline that involves querying a database, calculating cardinalities, flattening data, and finally outputting results in specific formats (e.g., `Int32`, `String`).













