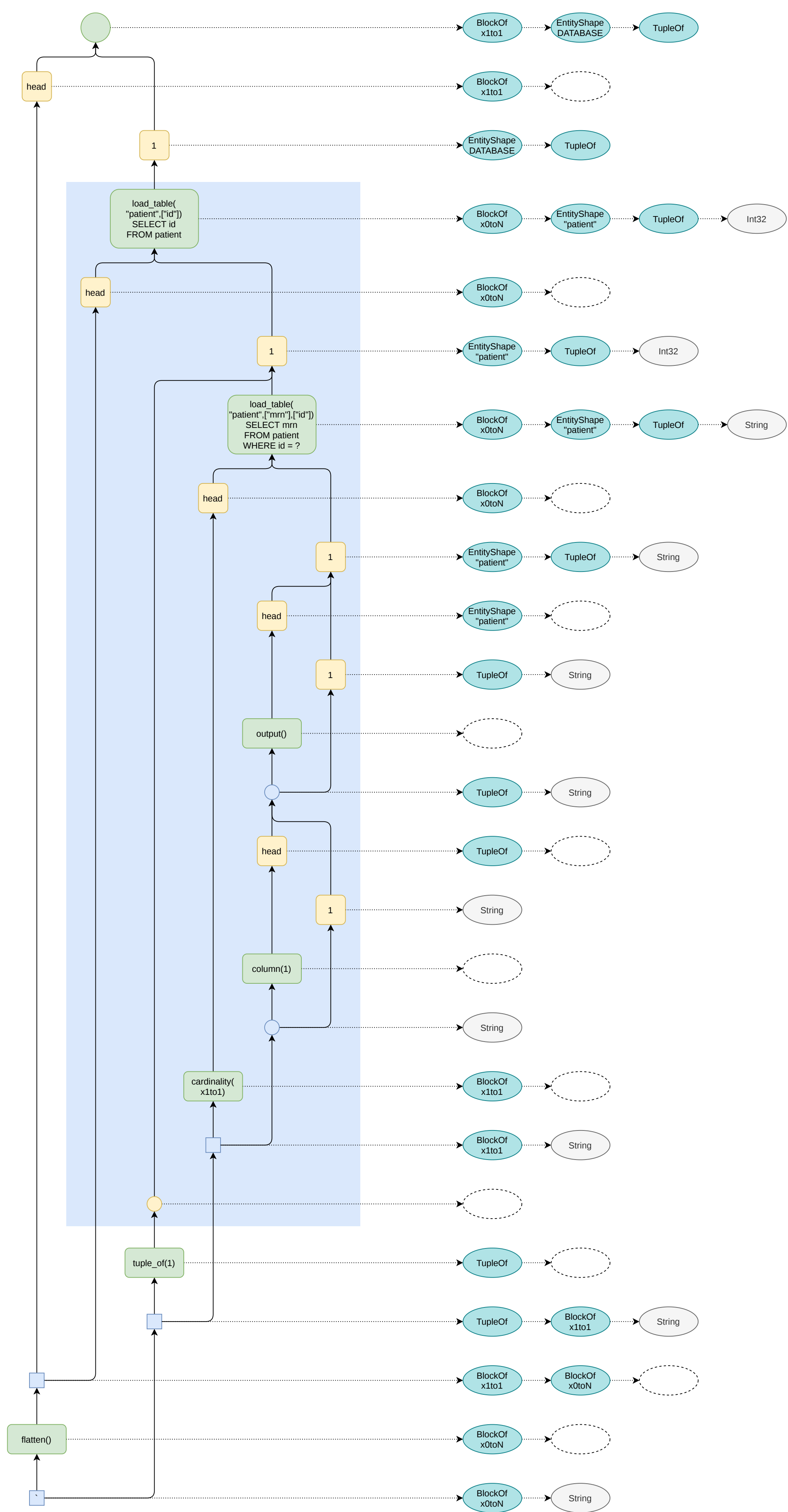


```
chain_of(
  with_elements(load_postgres_table(("public", "patient"), ["id"], [Int32])),
  flatten())
```



```
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 computational graph for a neural network, showing a sequence of operations from input to output. The graph is divided into a blue-shaded region and a red-shaded region.

**Blue-shaded region (Main Processing):**

- Input:** A green circle node at the top left.
- Initial Operations:** The input flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'load\_table("patient",["id"]) SELECT id FROM patient' node.
- Intermediate Operations:** The output of the 'load\_table' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'load\_table("patient",["mn"],["id"]) SELECT mn FROM patient WHERE id = ?' node.
- Cardinality and Flattening:** The output of the second 'load\_table' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'cardinality(x1to1)' node. The output of the 'cardinality' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'flatten()' node.
- Output:** The output of the 'flatten()' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'output()' node.

**Red-shaded region (Final Output):**

- Input:** A green circle node at the top left.
- Initial Operations:** The input flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'load\_table("patient",["mn"],["id"]) SELECT mn FROM patient WHERE id = ?' node.
- Intermediate Operations:** The output of the 'load\_table' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'cardinality(x1to1)' node. The output of the 'cardinality' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'flatten()' node.
- Output:** The output of the 'flatten()' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'output()' node.

**Final Output:** The output of the 'output()' node flows into a yellow 'head' node, then a yellow '1' node, and finally a green 'String' node.

**Connections and Data Flow:**

- The graph shows a complex network of connections between nodes, including a large blue-shaded region and a red-shaded region.
- The graph includes various nodes such as 'load\_table', 'cardinality', 'flatten', 'column', 'output', 'head', '1', 'BlockOf', 'EntityShape', and 'TupleOf'.
- The graph is a directed acyclic graph (DAG) representing the flow of data and operations.

















