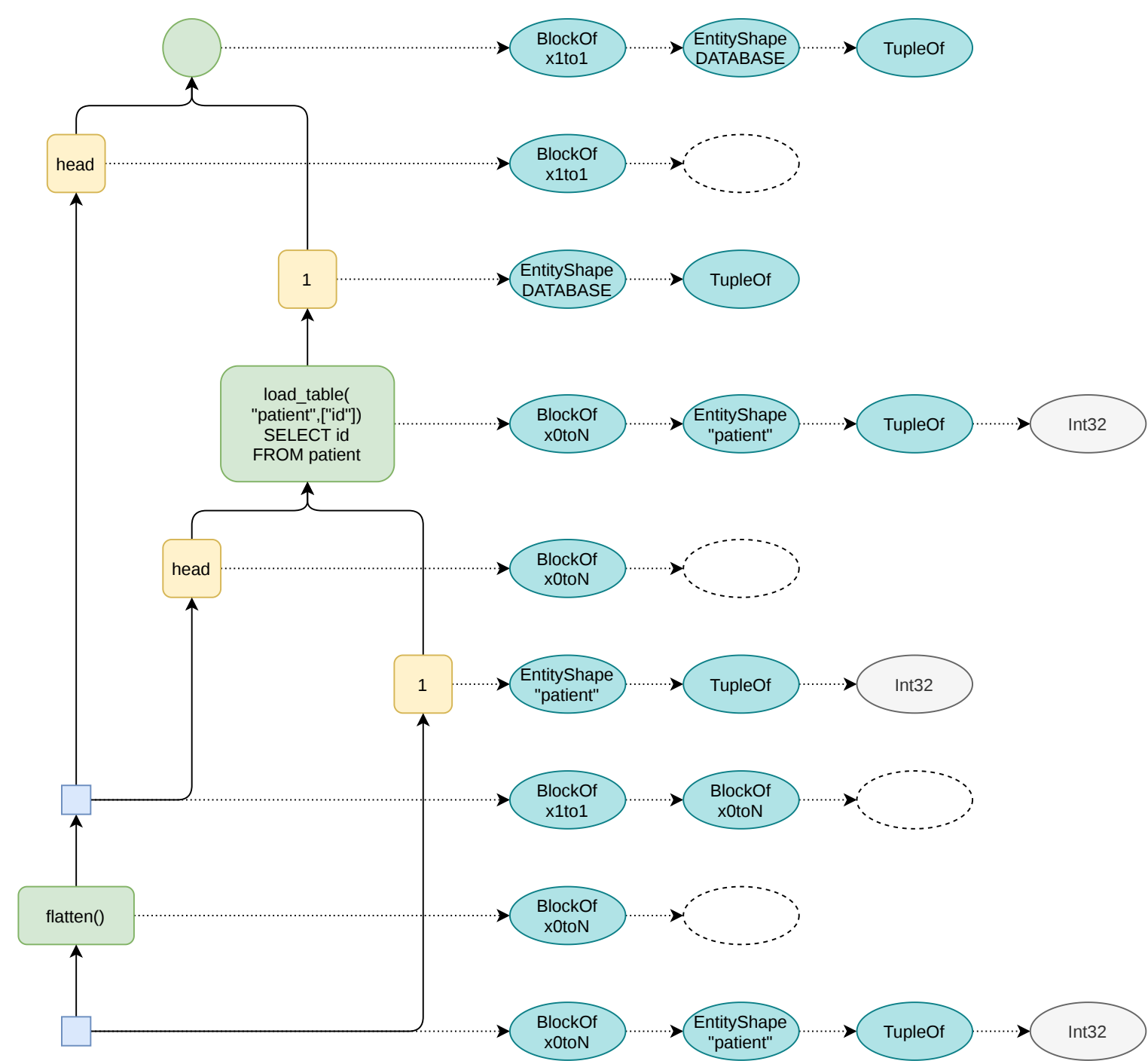


```
chain_of(
  with_elements(load_postgres_table(("public", "patient"), ["id"], [Int32])),
  flatten())
```



The diagram illustrates the transformation of a SQL query into a dataflow graph. The query is: `SELECT min(id) FROM patient WHERE id = ?`. The graph shows the flow of data from the query to the final output, which is a `String`.

The graph consists of several components:

- Query Execution:** The query is executed by a `load_table` node, which returns a `BlockOf x0toN` of `EntityShape "patient"` tuples.
- Cardinality Calculation:** A `cardinality(x1to1)` node calculates the cardinality of the result, returning a `BlockOf x1to1` of `EntityShape "patient"` tuples.
- Flattening:** The result is flattened by a `flatten()` node, returning a `BlockOf x0toN` of `BlockOf x1to1` of `EntityShape "patient"` tuples.
- Output:** The final result is output by an `output()` node, returning a `BlockOf x0toN` of `EntityShape "patient"` tuples.
- Column Selection:** A `column(1)` node selects the first column of the result, returning a `BlockOf x0toN` of `String` values.

The final output is a `String`.







```
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))))
```



















