

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



The diagram illustrates the execution of a query plan for a database query. The query is: `load_table('patient', ['id']) SELECT id FROM patient`. The diagram shows the flow of data through various operators and their corresponding data types.

Query Plan Structure:

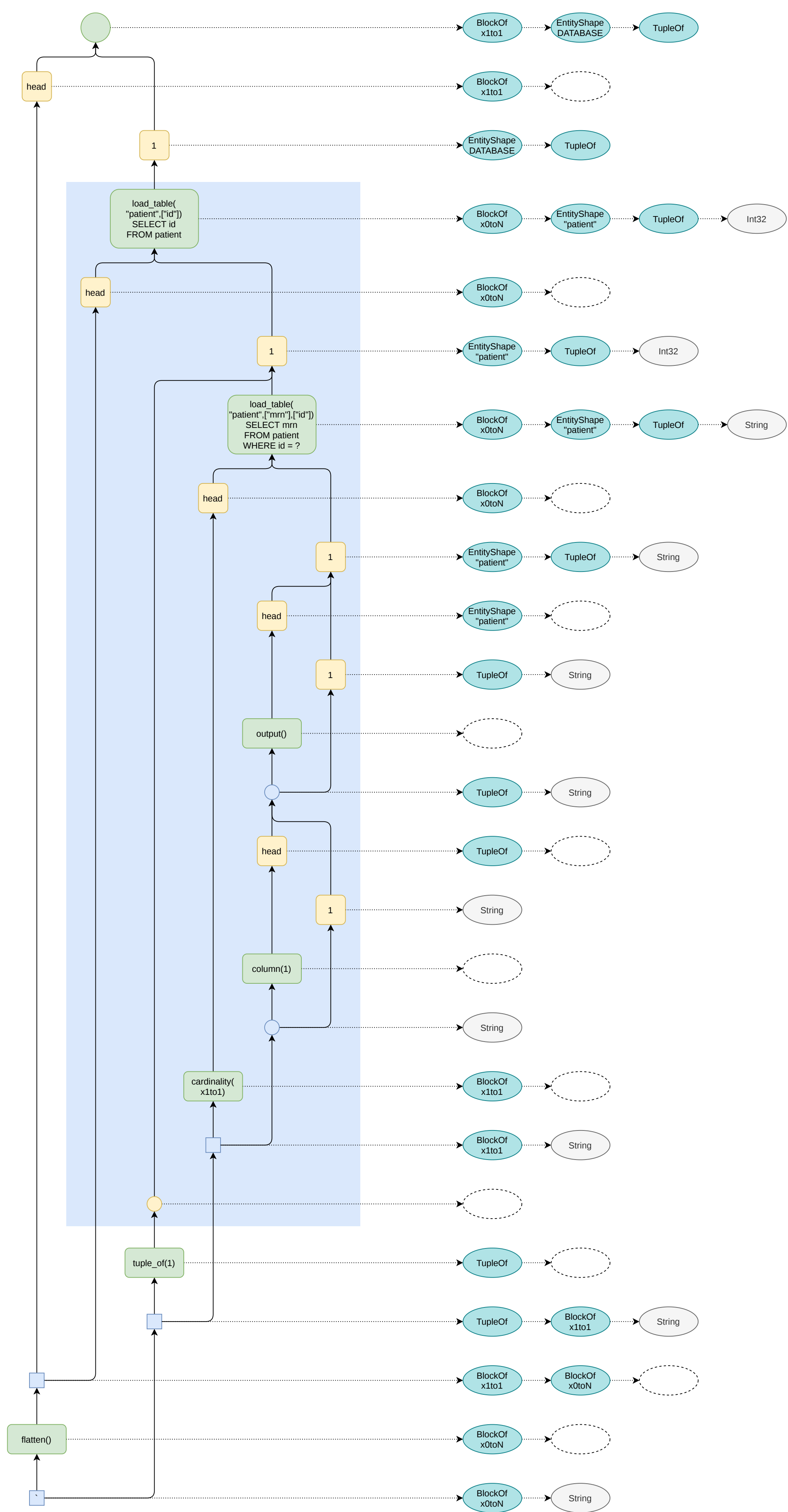
- Outer Query:** `load_table('patient', ['id']) SELECT id FROM patient`. This query is executed by a `head` operator, which outputs a `BlockOf x1to1` (a `TupleOf` containing an `EntityShape 'patient'` and a `TupleOf` containing an `Int32`).
- Inner Query:** `load_table('patient', ['min'], ['id']) SELECT min FROM patient WHERE id = ?`. This query is executed by a `head` operator, which outputs a `BlockOf x0toN` (a `TupleOf` containing an `EntityShape 'patient'` and a `TupleOf` containing a `String`).
- Cardinality Calculation:** A `cardinality(x1to1)` operator is used to calculate the cardinality of the inner query's output. It outputs a `BlockOf x1to1` (a `TupleOf` containing a `String`).
- Flattening:** The results of the inner query and the cardinality calculation are flattened using `flatten()` operators. The inner query's output is flattened into a `BlockOf x0toN` (a `TupleOf` containing a `String`).
- Output:** The final result is an `output()` operator that outputs a `TupleOf` containing a `String`.

Data Flow and Types:

- The `head` operator for the inner query outputs a `BlockOf x0toN` (a `TupleOf` containing an `EntityShape 'patient'` and a `TupleOf` containing a `String`).
- The `cardinality(x1to1)` operator outputs a `BlockOf x1to1` (a `TupleOf` containing a `String`).
- The `flatten()` operator for the inner query outputs a `BlockOf x0toN` (a `TupleOf` containing a `String`).
- The `output()` operator outputs a `TupleOf` containing a `String`.

The diagram uses a color-coded system to represent different data types and operators:

- Blue:** Operators (e.g., `head`, `load_table`, `cardinality`, `flatten`, `output`).
- Green:** Data types (e.g., `BlockOf`, `TupleOf`, `EntityShape`, `Int32`, `String`).
- Red:** Intermediate results (e.g., `BlockOf x1to1`, `BlockOf x0toN`).
- Yellow:** Final output (e.g., `TupleOf`).







```
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))))
```











