

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



The diagram illustrates the mapping between a query plan and its corresponding data types. The left side shows the query plan, and the right side shows the data types.

Query Plan (Left Side):

- load_table("patient", ["id"]) SELECT id FROM patient**: A green box representing a table load operation.
- cardinality(x1to1)**: A green box representing a cardinality operation.
- flatten()**: A green box representing a flatten operation.
- output()**: A green box representing an output operation.
- column(1)**: A green box representing a column selection operation.
- head**: Yellow boxes representing the head of a list or tuple.
- 1**: Yellow boxes representing a constant value of 1.
- Blue Shaded Area**: A large blue rectangle highlighting a specific part of the query plan.

Data Types (Right Side):

- BlockOf x1to1**: A teal oval representing a block of data.
- EntityShape DATABASE**: A teal oval representing an entity shape.
- TupleOf**: A teal oval representing a tuple.
- Int32**: A grey oval representing an integer.
- String**: A grey oval representing a string.

Arrows and Mappings:

- Arrows point from the query plan nodes to the data types, indicating the mapping.
- For example, the **load_table** node maps to **BlockOf x1to1**, **EntityShape DATABASE**, and **TupleOf**.
- The **cardinality** node maps to **BlockOf x0toN**.
- The **flatten** node maps to **BlockOf x1to1** and **BlockOf x0toN**.
- The **output** node maps to **TupleOf**.
- The **column** node maps to **String**.
- The **head** and **1** nodes map to various data types, including **BlockOf**, **EntityShape**, and **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))))
```















