

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



The diagram illustrates the execution of a SQL query, showing the flow of data and the corresponding data structures. The left side represents the query plan, and the right side represents the data structures.

**Query Plan (Left Side):**

- load\_table("patient", ["id"]) SELECT id FROM patient**: This node is highlighted in a light blue box. It has a **head** node and a **1** node.
- load\_table("patient", ["mrm"], ["id"]) SELECT mrm FROM patient WHERE id = ?**: This node is also highlighted in a light blue box. It has a **head** node and a **1** node.
- cardinality(x1to1)**: This node is highlighted in a light red box. It has a **head** node and a **1** node.
- flatten()**: Two instances of this node are shown, each with a **head** node and a **1** node.
- output()**: This node has a **head** node and a **1** node.
- column(1)**: This node has a **head** node and a **1** node.

**Data Structures (Right Side):**

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

The diagram shows the flow of data from the query plan to the data structures. The **load\_table** nodes are connected to **BlockOf** and **EntityShape** nodes. The **cardinality** node is connected to **BlockOf** and **TupleOf** nodes. The **flatten** nodes are connected to **BlockOf** and **TupleOf** nodes. The **output** node is connected to **TupleOf** and **String** nodes. The **column** node is connected to **BlockOf** and **String** nodes.







```
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))))
```



















