

```
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 the mapping of a query plan to a relational algebra expression. The query plan on the left is a directed graph with nodes representing operations and data. The relational algebra expression on the right is a corresponding graph using standard relational algebra constructs.

**Query Plan (Left):**

- Input:** A blue square node at the bottom left.
- flatten():** A green rounded rectangle node.
- Join:** A blue circle node with a plus sign, receiving input from the **flatten()** node and a **column(1)** node.
- column(1):** A green rounded rectangle node.
- head:** A yellow rounded rectangle node.
- output():** A green rounded rectangle node.
- Join:** A blue circle node with a plus sign, receiving input from the **output()** node and a **1** node.
- 1:** A yellow rounded rectangle node.
- cardinality(x1to1):** A green rounded rectangle node.
- Join:** A blue circle node with a plus sign, receiving input from the **cardinality(x1to1)** node and a **1** node.
- head:** A yellow rounded rectangle node.
- load\_table("patient",["min"],["id"]):** A green rounded rectangle node.
- 1:** A yellow rounded rectangle node.
- head:** A yellow rounded rectangle node.
- load\_table("patient",["id"]):** A green rounded rectangle node.
- 1:** A yellow rounded rectangle node.
- head:** A yellow rounded rectangle node.
- flatten():** A green rounded rectangle node.
- Join:** A blue circle node with a plus sign, receiving input from the **flatten()** node and a **1** node.
- head:** A yellow rounded rectangle node.
- 1:** A yellow rounded rectangle node.
- Output:** A green circle node at the top.

**Relational Algebra Expression (Right):**

- BlockOf x1to1** and **EntityShape DATABASE** lead to **TupleOf**.
- BlockOf x1to1** leads to a dashed oval.
- EntityShape DATABASE** leads to **TupleOf**.
- BlockOf x0toN** and **EntityShape "patient"** lead to **TupleOf**, which then leads to **Int32**.
- BlockOf x0toN** leads to a dashed oval.
- EntityShape "patient"** leads to **TupleOf**, which then leads to **Int32**.
- BlockOf x0toN** and **EntityShape "patient"** lead to **TupleOf**, which then leads to **String**.
- BlockOf x0toN** leads to a dashed oval.
- EntityShape "patient"** leads to **TupleOf**, which then leads to **String**.
- BlockOf x1to1** leads to a dashed oval.
- BlockOf x0toN** and **BlockOf x1to1** lead to a dashed oval.
- BlockOf x0toN** leads to a dashed oval.
- BlockOf x1to1** and **BlockOf x0toN** lead to a dashed oval.
- BlockOf x0toN** leads to a dashed oval.
- EntityShape "patient"** leads to a dashed oval.
- TupleOf** leads to **String**.
- Join:** A blue circle node with a plus sign, receiving input from a **TupleOf** node and a **1** node.
- 1** leads to **String**.
- Join:** A blue circle node with a plus sign, receiving input from a **1** node and a **1** node.
- 1** leads to **String**.
- Join:** A blue circle node with a plus sign, receiving input from a **1** node and a **1** node.
- 1** leads to **String**.
- BlockOf x0toN** leads to **String**.

















