

```
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 SQL query to a graph structure. The left side shows the query logic, and the right side shows the resulting data structure.

SQL Query:

```

load_table('patient', ['id'])
SELECT id
FROM patient;

load_table('patient', ['mrm'], ['id'])
SELECT min
FROM patient
WHERE id = ?;

cardinality(x1to1);

flatten();

output();

column(1);

```

Graph Structure Mapping:

- load_table('patient', ['id']) SELECT id FROM patient;**
 - BlockOf x1to1 → EntityShape DATABASE → TupleOf
 - BlockOf x1to1 → (empty dashed oval)
 - EntityShape DATABASE → TupleOf
 - BlockOf x0toN → EntityShape "patient" → TupleOf → Int32
 - BlockOf x0toN → (empty dashed oval)
 - EntityShape "patient" → TupleOf → Int32
- load_table('patient', ['mrm'], ['id']) SELECT min FROM patient WHERE id = ?;**
 - BlockOf x0toN → EntityShape "patient" → TupleOf → String
 - BlockOf x0toN → (empty dashed oval)
 - EntityShape "patient" → TupleOf → String
- cardinality(x1to1);**
 - BlockOf x1to1 → (empty dashed oval)
 - BlockOf x0toN → BlockOf x1to1 → (empty dashed oval)
 - BlockOf x0toN → (empty dashed oval)
- flatten();**
 - BlockOf x1to1 → BlockOf x0toN → (empty dashed oval)
 - BlockOf x0toN → (empty dashed oval)
- output();**
 - EntityShape "patient" → (empty dashed oval)
 - TupleOf → String
 - (empty dashed oval) → TupleOf → String
 - TupleOf → (empty dashed oval)
- column(1);**
 - String → (empty dashed oval)
 - (empty dashed oval) → String
 - BlockOf x0toN → String













