

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



[illegible]







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. The mapping is as follows:

- Query Plan Nodes:**
 - `flatten()` (green box)
 - `cardinality(x1to1)` (green box)
 - `load_table("patient",["id"]) SELECT id FROM patient` (green box)
 - `load_table("patient",["min","id"]) SELECT min FROM patient WHERE id = ?` (green box)
 - `output()` (green box)
 - `column(1)` (green box)
 - `head` (yellow box)
 - `1` (yellow box)
 - `BlockOf x1to1` (blue box)
 - `BlockOf x0toN` (blue box)
 - `EntityShape "patient"` (blue box)
 - `TupleOf` (blue box)
 - `Int32` (grey oval)
 - `String` (grey oval)
- Relational Algebra Expression Nodes:**
 - `BlockOf x1to1` (blue box)
 - `EntityShape "patient"` (blue box)
 - `TupleOf` (blue box)
 - `Int32` (grey oval)
 - `String` (grey oval)
- Mapping:**
 - `flatten()` maps to `BlockOf x1to1`.
 - `cardinality(x1to1)` maps to `BlockOf x0toN`.
 - `load_table("patient",["id"]) SELECT id FROM patient` maps to `EntityShape "patient"`.
 - `load_table("patient",["min","id"]) SELECT min FROM patient WHERE id = ?` maps to `TupleOf`.
 - `output()` maps to `Int32`.
 - `column(1)` maps to `String`.
 - `head` maps to `BlockOf x1to1`.
 - `1` maps to `BlockOf x0toN`.
 - `BlockOf x1to1` maps to `BlockOf x0toN`.
 - `BlockOf x0toN` maps to `BlockOf x1to1`.
 - `EntityShape "patient"` maps to `TupleOf`.
 - `TupleOf` maps to `Int32`.
 - `Int32` maps to `String`.
 - `String` maps to `String`.













