

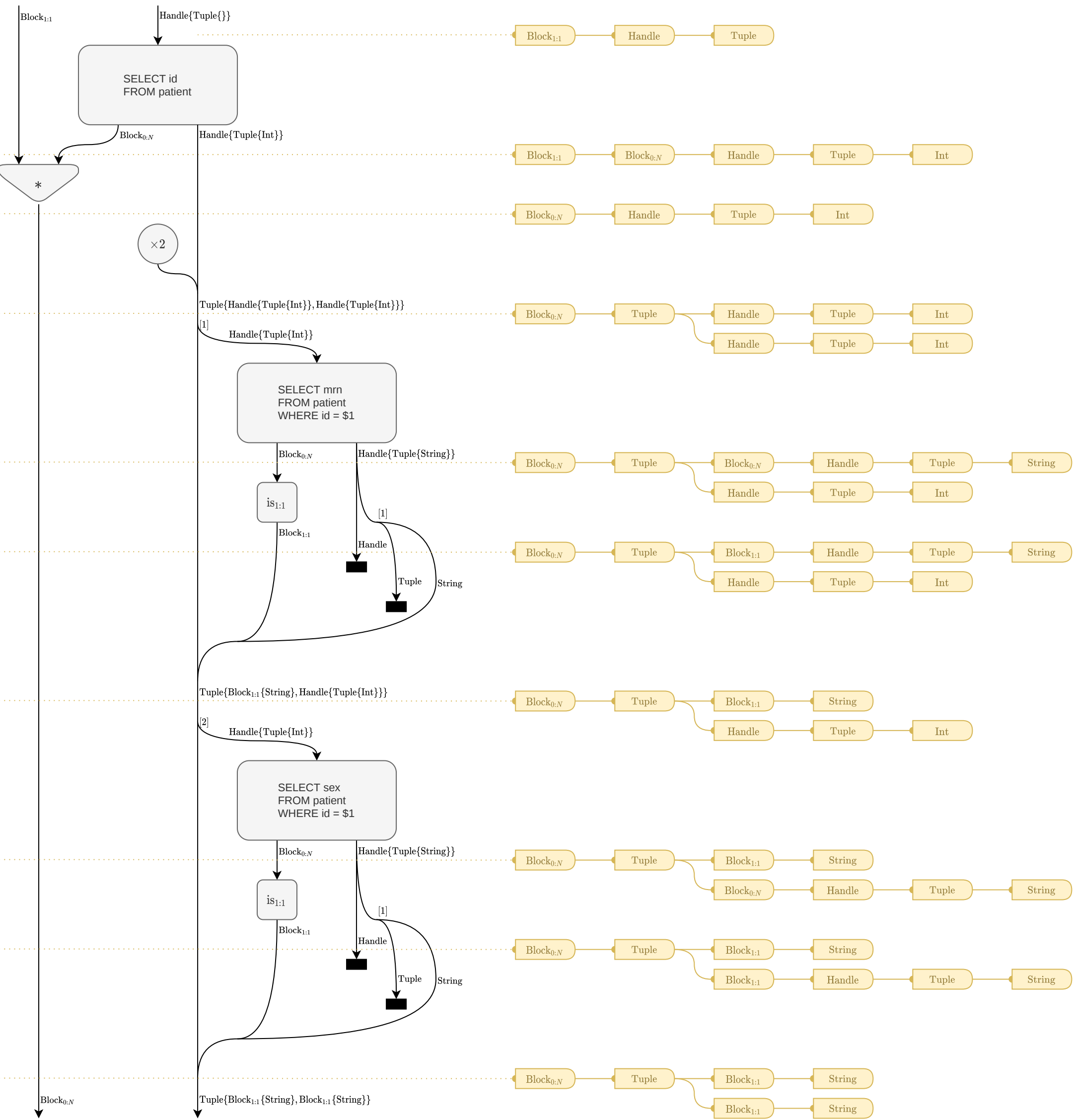
@query db patient

chain\_of(with\_elements(load\_postgres\_table(("public", "patient")), ["id"], [Int32])), flatten())



```
@query db patient{mrn,sex}
```

```
chain_of(with_elements(load_postgres_table("public", "patient"), ["id"], [Int32]),
  flatten(),
  with_elements(tuple_of:(mrn => chain_of(load_postgres_table("public", "patient"), ["mrn"], [String], ["id"],
    block_cardinality(x1tol),
    with_elements(chain_of(output(), column(1))),
    :sex => chain_of(load_postgres_table("public", "patient"), ["sex"], [String], ["id"],
      block_cardinality(x1tol),
      with_elements(chain_of(output(), column(1)))))))
```



SQL Pushdown

@query db patient{mrn,sex}

```
chain_of(with_elements(load_postgres_table(("public", "patient"), ["mrn", "sex"], [String, String])),
flatten(),
with_elements(output()))
```



```
struct EvalNode
  p::Pipeline
  i::NodeRef
end

Nothing

struct SlotNode
  n::NodeRef
  k::Int
end

struct HeadNode
  n::NodeRef
end

struct JoinNode
  head::NodeRef
  slots::Vector{NodeRef}
end

struct InputNode
end

mutable struct NodeRef
  node::Union{Nothing,Node}
end
```

