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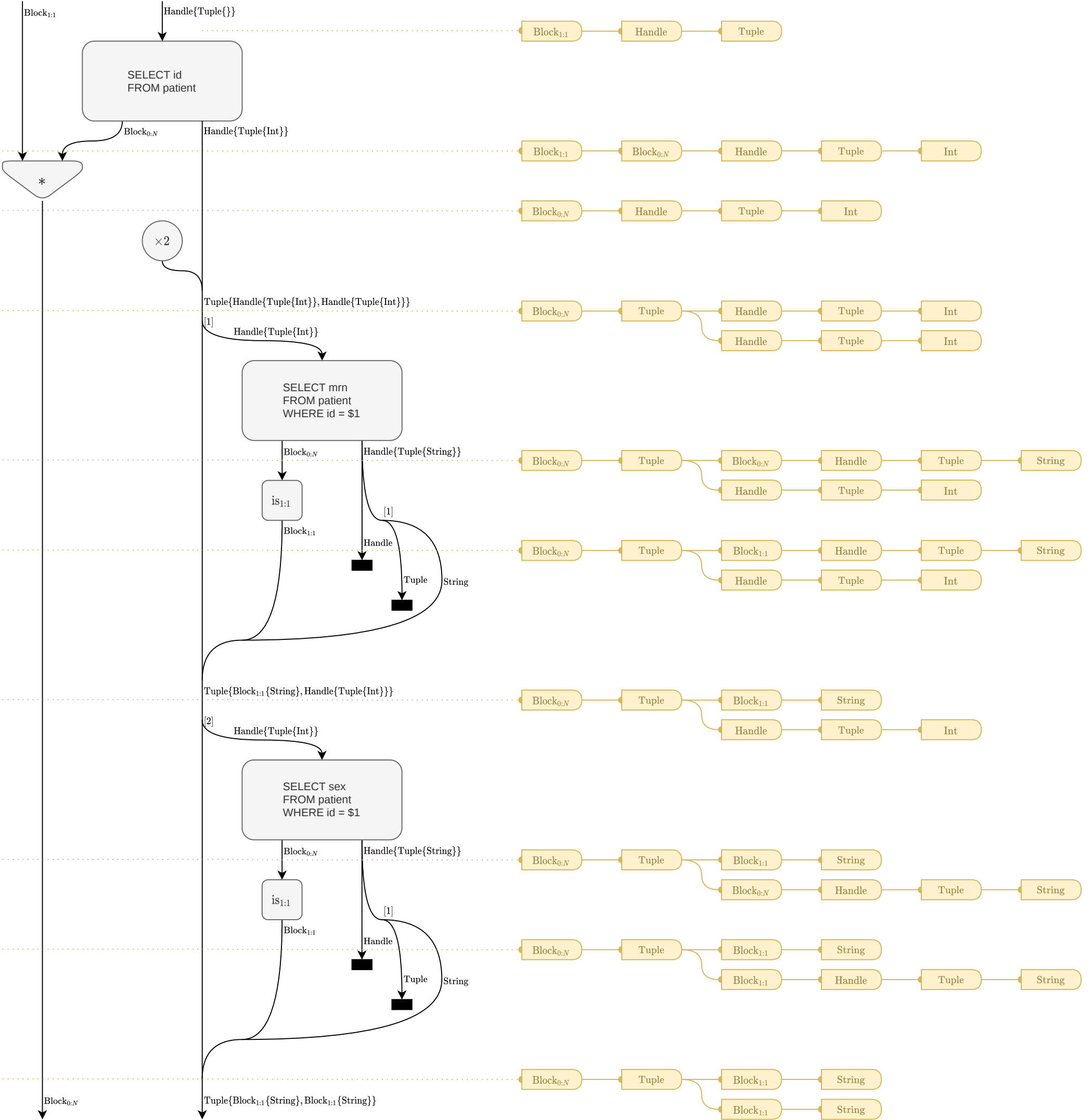
```
chain_of(with_elements(load_postgres_table(("public", "patient"), ["id"], [Int32])), flatten())
```


$\text{Block}_{1:1}$ — Handle — $\text{Tuple}\{\}$

db

$\text{Block}_{1:1}$ · Handle · Tuple

$\text{Block}_{1:1}$

$\text{Handle}\{\text{Tuple}\{\}\}$

SELECT id
FROM patient

$\text{Block}_{0:N}$

*

$\text{Block}_{0:N}$

$\text{Handle}\{\text{Tuple}\{\text{Int}\}\}$

$\text{Block}_{1:1}$ · Handle · Tuple

$\text{Block}_{1:1}$ · $\text{Block}_{0:N}$ · Handle · Tuple · Int

$\text{Block}_{0:N}$ · Handle · Tuple · Int

@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(x1to1),
                                               with_elements(chain_of(output(), column(1)))),
                      :sex => chain_of(load_postgres_table(("public", "patient"), ["sex"], [String], ["id"]),
                                               block_cardinality(x1to1),
                                               with_elements(chain_of(output(), column(1))))))))
```


Block$_{1:1}$

Handle{Tuple{}}

Block$_{1:1}$ — Handle — Tuple

SELECT id
FROM patient

Block$_{0:N}$ Handle{Tuple{Int}}

Block$_{1:1}$ — Block$_{0:N}$ — Handle — Tuple — Int

*

Block$_{0:N}$ — Handle — Tuple — Int

×2

Tuple{Handle{Tuple{Int}},Handle{Tuple{Int}}}

[1]   Handle{Tuple{Int}}

Block$_{0:N}$ — Tuple — Handle — Tuple — Int
                       Handle — Tuple — Int

SELECT mrn
FROM patient
WHERE id = $1

Block$_{0:N}$ Handle{Tuple{String}}

Block$_{0:N}$ — Tuple — Block$_{0:N}$ — Handle — Tuple — String
                       Handle — Tuple — Int

is$_{1:1}$

[1]

Block$_{1:1}$ Handle

Block$_{0:N}$ — Tuple — Block$_{1:1}$ — Handle — Tuple — String
                       Handle — Tuple — Int

Tuple   String

Tuple{Block$_{1:1}${String},Handle{Tuple{Int}}}

Block$_{0:N}$ — Tuple — Block$_{1:1}$ — String
                       Handle — Tuple — Int

[2]   Handle{Tuple{Int}}

SELECT sex
FROM patient
WHERE id = $1

Block$_{0:N}$ Handle{Tuple{String}}

Block$_{0:N}$ — Tuple — Block$_{1:1}$ — String
                       Block$_{0:N}$ — Handle — Tuple — String

is$_{1:1}$

[1]

Block$_{1:1}$ Handle

Block$_{0:N}$ — Tuple — Block$_{1:1}$ — String
                       Block$_{1:1}$ — Handle — Tuple — String

Tuple   String

Block$_{0:N}$

Tuple{Block$_{1:1}${String},Block$_{1:1}${String}}

Block$_{0:N}$ — Tuple — Block$_{1:1}$ — String
                       Block$_{1:1}$ — String

SQL Pushdown

@query db patient{mrn,sex}

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

$\text{Block}_{1:1}$

$\text{Handle}\{\text{Tuple}\{\}\}$

SELECT mrn, sex
FROM patient

$\text{Block}_{0:N}$

$\text{Handle}\{\text{Tuple}\{\text{String}, \text{String}\}\}$

*

Handle

$\text{Block}_{0:N}$

$\text{Tuple}\{\text{String}, \text{String}\}$

| $\text{Block}_{1:1}$ | Handle | Tuple |

| $\text{Block}_{1:1}$ | $\text{Block}_{0:N}$ | Handle | Tuple | String |
| | | | | String |

| $\text{Block}_{0:N}$ | Handle | Tuple | String |
| | | | String |

| $\text{Block}_{0:N}$ | Tuple | String |
| | | String |