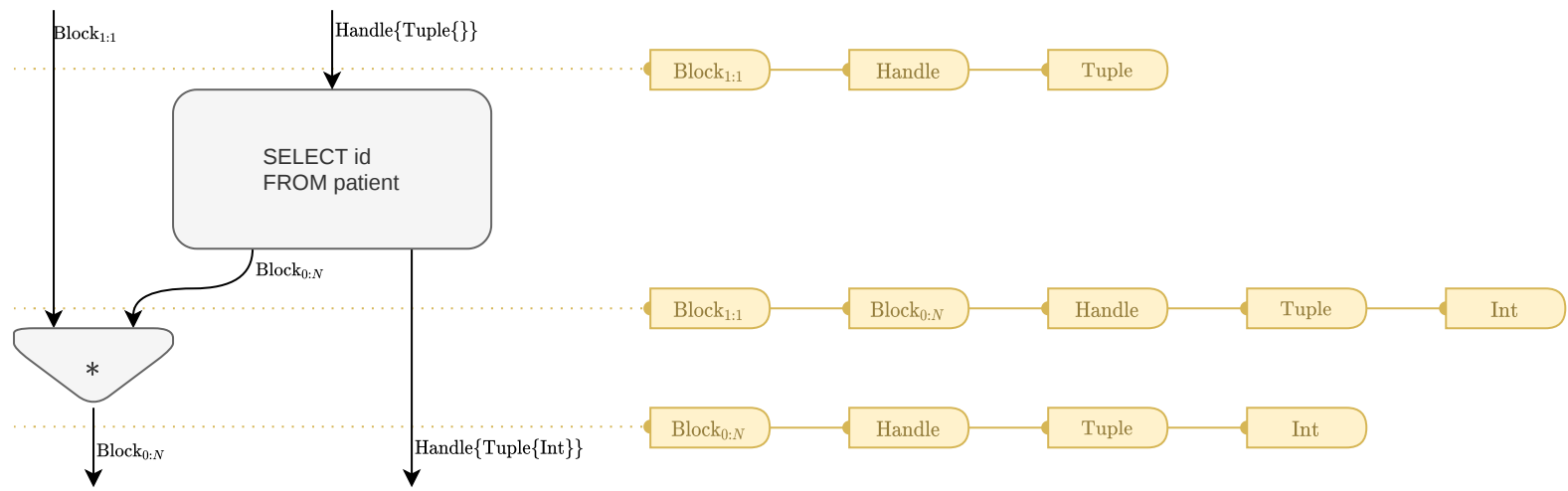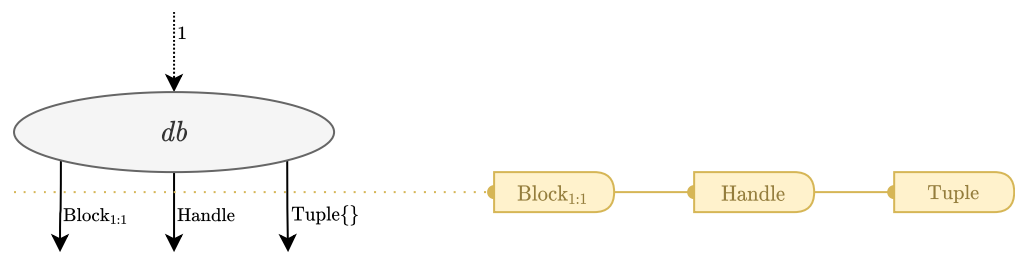**@query db patient**

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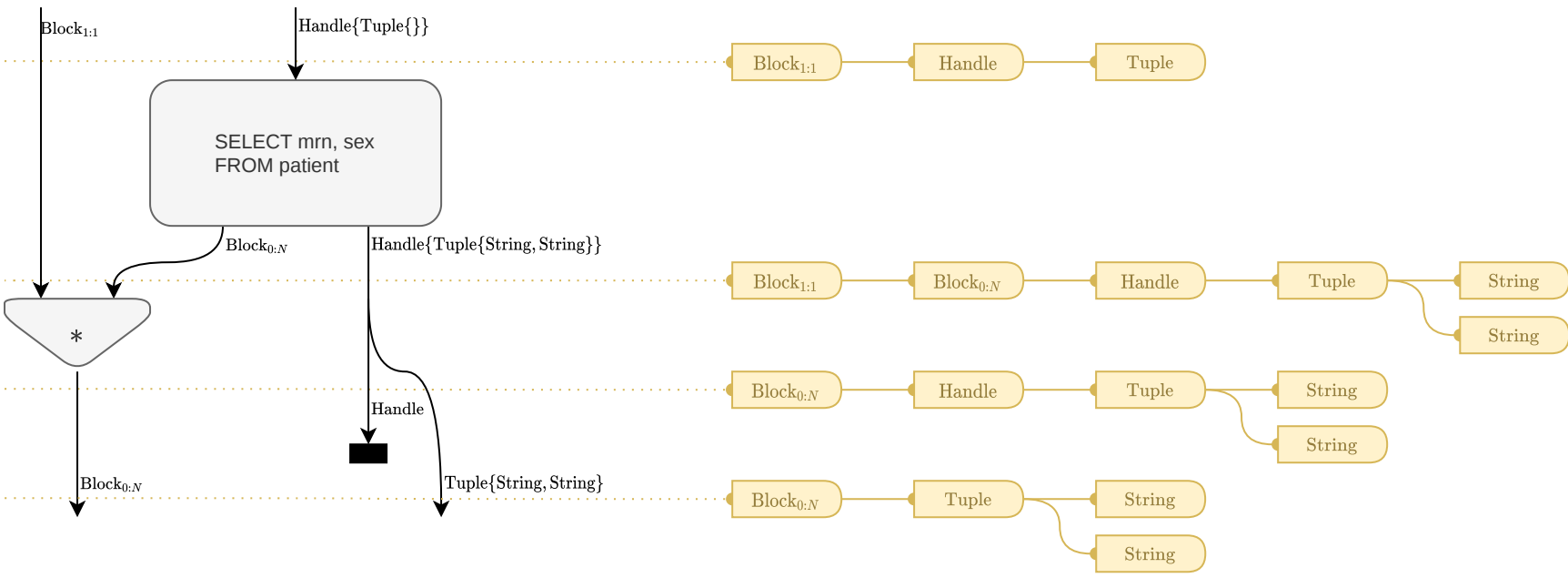
@query db patient{mrn,sex}

```
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(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)))))))))
```

SQL Pushdown

@query db patient{mrn,sex}

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

$Block_{1,1}$ $Handle\{Tuple\{\}\}$

| $Block_{1,1}$ | $Handle$ | $Tuple$ |

SELECT mrn, sex
FROM patient

$Block_{R,N}$ $Handle\{Tuple\{String, String\}\}$

| $Block_{1,1}$ | $Block_{R,N}$ | $Handle$ | $Tuple$ | $String$ |
| | | | | $String$ |

$Handle$

| $Block_{R,N}$ | $Handle$ | $Tuple$ | $String$ |
| | | | $String$ |

$*$

$Block_{R,N}$ $Tuple\{String, String\}$

| $Block_{R,N}$ | $Tuple$ | $String$ |
| | | $String$ |

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

p

Nothing

```
struct SlotNode
    n::NodeRef
    k::Int
end
```

n

```
struct HeadNode
    n::NodeRef
end
```

head

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

```
struct InputNode
end
```

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