



SELECT "Hello World!"

sql\_select("Hello World!")

sql\_query() |> sql\_select("Hello World!")



SELECT p.mrn FROM patient p

p = sql\_alias("patient")

sql\_join(p) |> sql\_select(p.mrn)

p = sql\_alias("patient")

p |> sql\_select(p.mrn)

(p = sql\_from("patient")) |> sql\_select(p.mrn)



SELECT p.mrn, e.date

FROM patient p

JOIN encounter e ON (p.id = e.patient\_id)

p = sql\_alias("patient")

e = sql\_alias("encounter")

sql\_from(p) |> sql\_join(e, p.id, == e.patient\_id) |> sql\_select(p.mrn, e.date)

p = sql\_alias(catalog["public"]["patient"])

e = sql\_alias(catalog["public"]["encounter"])

sql\_from(p) |> sql\_join(e, autojoin=p) |> sql\_select(p.mrn, e.date)

p = sql\_alias("patient")

e = sql\_alias("encounter")

p |> sql\_join(e, p.id, == e.patient\_id) |> sql\_select(p.mrn, e.date)

p = sql\_alias("patient")

e = sql\_alias("encounter")

sql\_from(p) |> sql\_join(e, p.id, == e.patient\_id) |> sql\_select(p.mrn) |> sql\_select(e.date)



SELECT p.mrn, e.date

FROM patient p

JOIN encounter e ON (p.id = e.patient\_id)

p = From("patient")

e = From("encounter")

j = Join(p, e, p.id, == e.patient\_id)

Select(j, p.mrn, e.date)

sql\_from((p = sql\_alias("patient")) |> sql\_join((e = sql\_alias("encounter")), p.id, == e.patient\_id) |> sql\_select(p.mrn, e.date))



SELECT p.sex, COUNT(p)

FROM patient p

GROUP BY p.sex

p = sql\_alias("patient")

g = sql\_from(p) |> sql\_group(sex = p.sex)

g |> sql\_select(g.sex, sql\_count(p))

p = From("patient")

g = Group(p, sex = p.sex)

Select(g, g.sex, Count(p))



SELECT p.mrn, COALESCE(g.n\_e, 0)

FROM patient p

LEFT JOIN (

SELECT e.patient\_id, COUNT(e) AS n\_e

FROM encounter e

GROUP BY e.patient\_id) g ON (p.id = g.patient\_id)

p = From("patient")

e = From("encounter")

g = Group(e, patient\_id = e.patient\_id)

j = LeftJoin(p, g, p.id, == g.patient\_id, omit\_if\_unused=true)

Select(j, p.mrn, Coalesce(Count(e), 0))

p = From("patient")

e = From("encounter")

g = Group(e, patient\_id = e.patient\_id)

gs = Select(g, patient\_id = g.patient\_id, n = Count(e))

j = LeftJoin(p, gs, p.id, == gs.patient\_id)

Select(j, p.mrn, Coalesce(gs.n, 0))

p = From("patient")

e = From("encounter")

g = Group(e, patient\_id = e.patient\_id, summarize=(; n = Count(e)))

j = LeftJoin(p, g, p.id, == g.patient\_id)

Select(j, p.mrn, Coalesce(g.n, 0))

SELECT p.mrn

FROM patient p

WHERE p.sex = 'male'

p = From("patient")

w = Where(p, p.sex, == "male")

Select(w, p.mrn)

p = From("patient", columns=["mrn", "sex"])

w = Where(p, Ref(1, 2), == "male", select=[Ref(1,1)])

Select(w, select=[Ref(1,1)])

patient\_tbl = Table("patient", [{"id", Int}, {"sex", String}, {"mrn", String}])

encounter\_tbl = Table("encounter", [{"id", Int}, {"patient\_id", Int}, {"date", Date}])

auto\_connect(patient\_tbl, encounter\_tbl, [{"id", "patient\_id"}])

p = From(patient\_tbl)

e = From(encounter\_tbl)

j = LeftJoin(p, e)

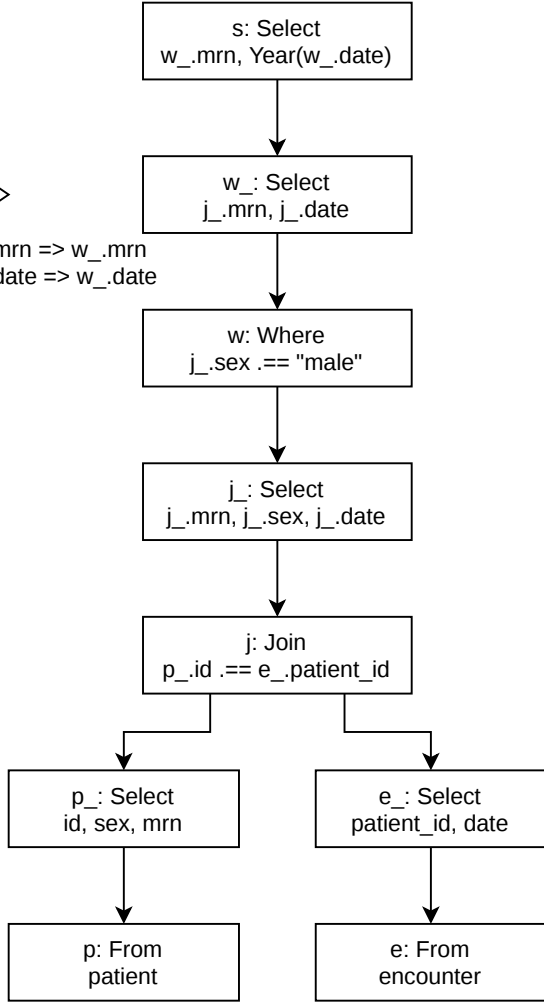
Select(j, p.mrn, e.date)



SELECT p.mrn, EXTRACT(YEAR FROM e.date)  
FROM patient p  
JOIN encounter e  
ON (p.id = e.patient\_id)  
WHERE p.sex = 'male'



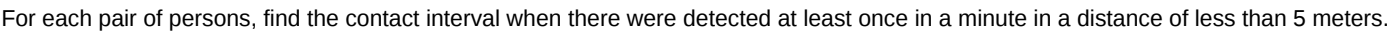
```
p = From(patient)
p_ = Select(p, id = Const(:id), _sex = Const(:sex), _mrn = Const(mrn))
e = From(encounter)
e_ = Select(e, _patient_id = Const(patient_id), _date = Const(:date))
j = Join(p_, e_, p_.id := e_.patient_id)
l_ = Select(j, _mrn = p_.mrn, _sex = p_.sex, _date = e_.date)
w = Where(l_, l_.sex := "male")
w_ = Select(w, mrn = l_.mrn, _date = l_.date)
s = Select(w_, mrn = w_.mrn, year = Year(w_.date))
```



normalize

p.mrn => w\_mrn  
e.date => w\_date

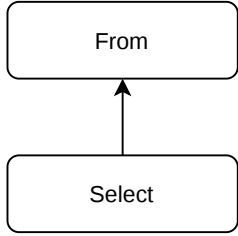
For each pair of persons, find the contact interval when there were detected at least once in a minute in a distance of less than 5 meters.



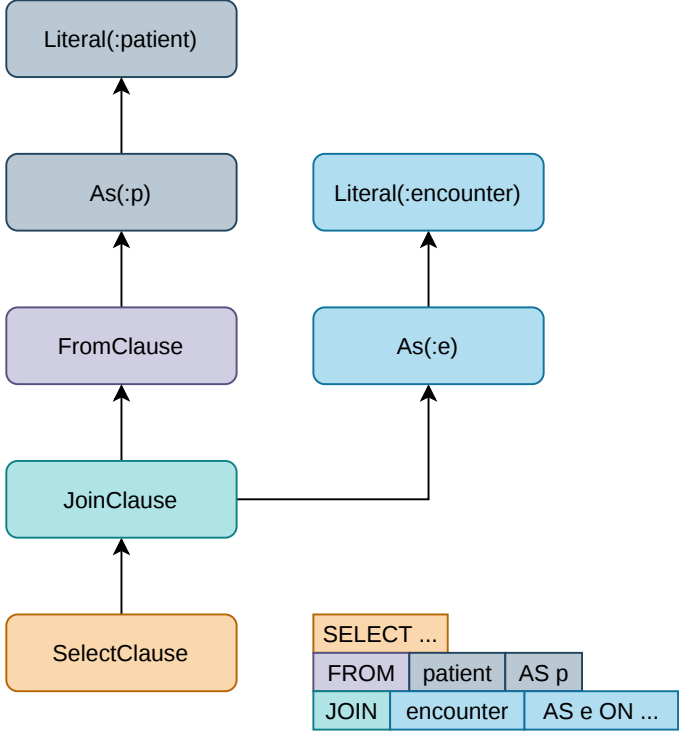
SELECT p.mrn, e.date  
FROM patient p  
JOIN encounter e ON (p.id = e.patient\_id)  
WHERE p.sex = 'male'  
ORDER BY e.date



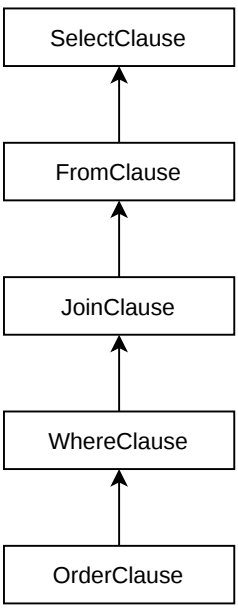
SELECT p.mrn  
FROM patient p



SELECT p.mrn, e.date  
FROM patient p  
JOIN encounter e  
ON (p.id = e.patient\_id)



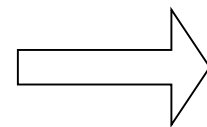
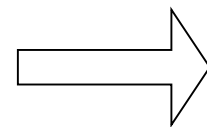
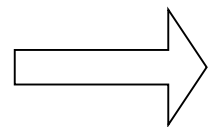
SELECT ...  
FROM patient AS p  
JOIN encounter AS e ON ...



Clause  
:SELECT

Clause  
:FROM

Clause  
:JOIN





```
WITH RECURSIVE X AS (  
  SELECT 1 AS N  
  UNION ALL  
  SELECT ...  
  ...  
  FROM X  
  ...  
  FROM X)
```







SELECT ... FROM ( SELECT ... ... ) AS ...



SELECT ... FROM ( SELECT ... ) AS ... WHERE ...



SELECT ... FROM ( SELECT ... FROM ... ) AS ... WHERE ...



SELECT ... FROM ( SELECT ... ... WHERE ... ) AS ... WHERE ...



SELECT ... FROM ( SELECT ... ... JOIN ... ) AS ... WHERE ...



