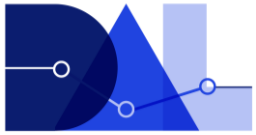
CEFET/RJ



ANÁLISE DE PLANOS DE EXECUÇÃO DE CONSULTAS NOS SBGD

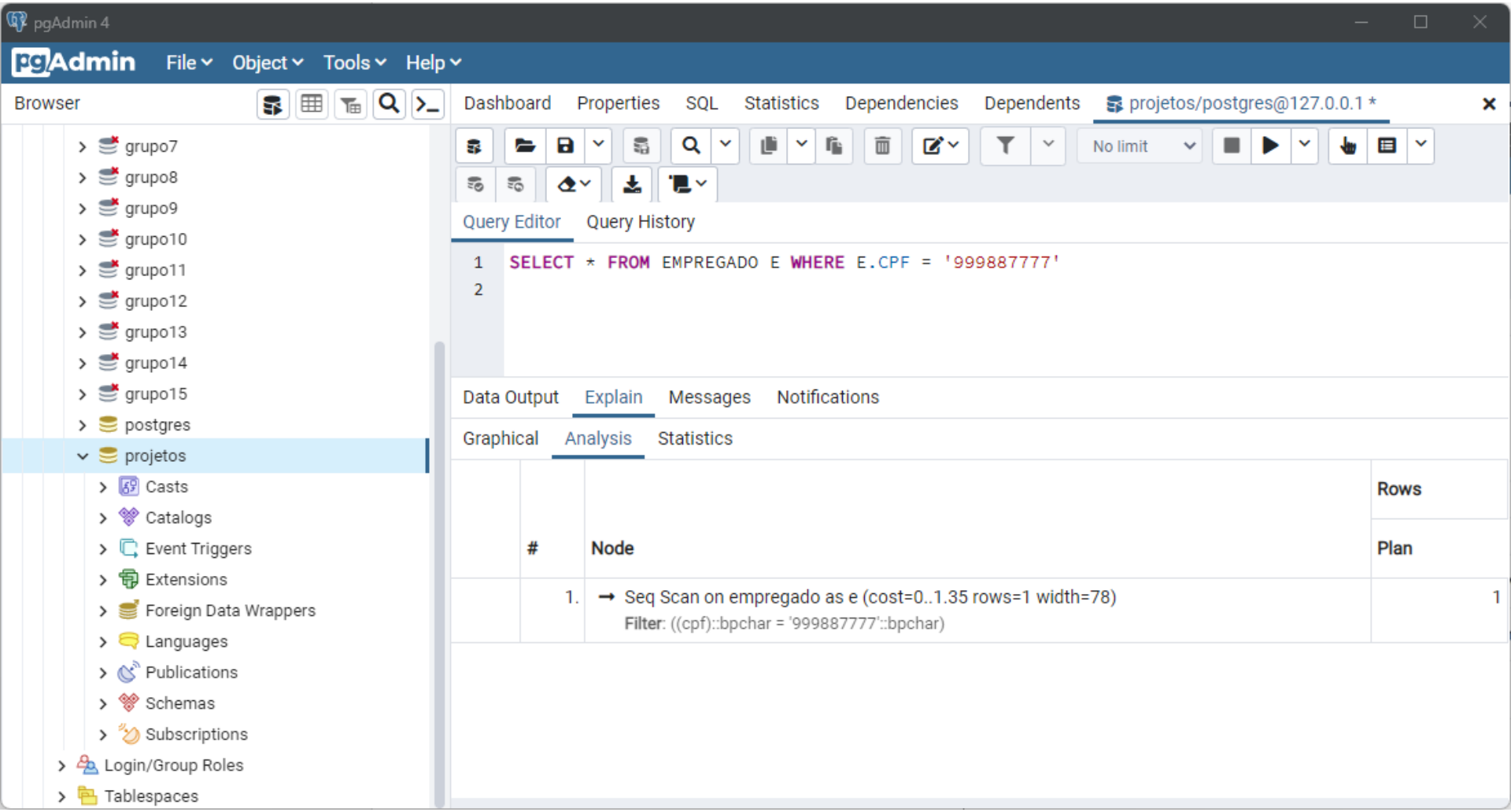
Eduardo Ogasawara
eogasawara@ieee.org
https://eic.cefet-rj.br/~eogasawara

Esquema de análise

- DEPARTAMENTO
  - PRIMARY KEY (DNUMERO)
- EMPREGADO
  - PRIMARY KEY (CPF)
  - FOREIGN KEY (GERENTECPF) REFERENCES EMPREGADO (CPF),
  - FOREIGN KEY (DNO) REFERENCES DEPARTAMENTO (DNUMERO)
- DEPT_LOCALIZACOES
  - PRIMARY KEY (DNUMERO,DLOCALIZACAO),
  - FOREIGN KEY (DNUMERO) REFERENCES DEPARTAMENTO (DNUMERO)
- PROJETO
  - PRIMARY KEY (PNUMERO)
  - FOREIGN KEY (DNUM) REFERENCES DEPARTAMENTO (DNUMERO)
- TRABALHA_EM
  - PRIMARY KEY (ECPF,PNO)
  - FOREIGN KEY (ECPF) REFERENCES EMPREGADO (CPF),
  - FOREIGN KEY (PNO) REFERENCES PROJETO (PNUMERO)
- DEPENDENTE
  - PRIMARY KEY (ECPF, NOME_DEPENDENTE),
  - FOREIGN KEY (ECPF) REFERENCES EMPREGADO (CPF)

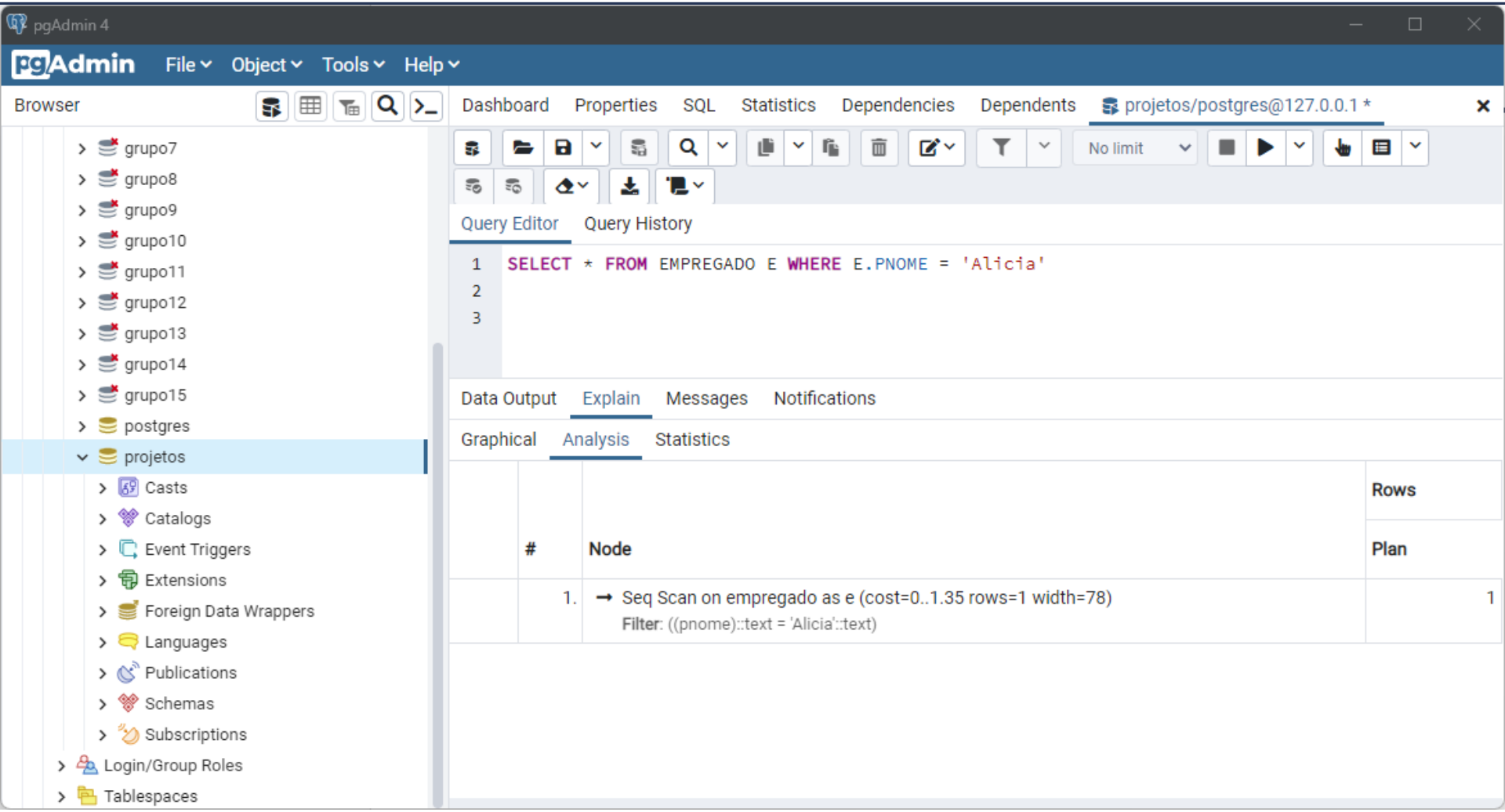
PostgreSQL

Consultas pela chave primária (empregado a partir do cpf)



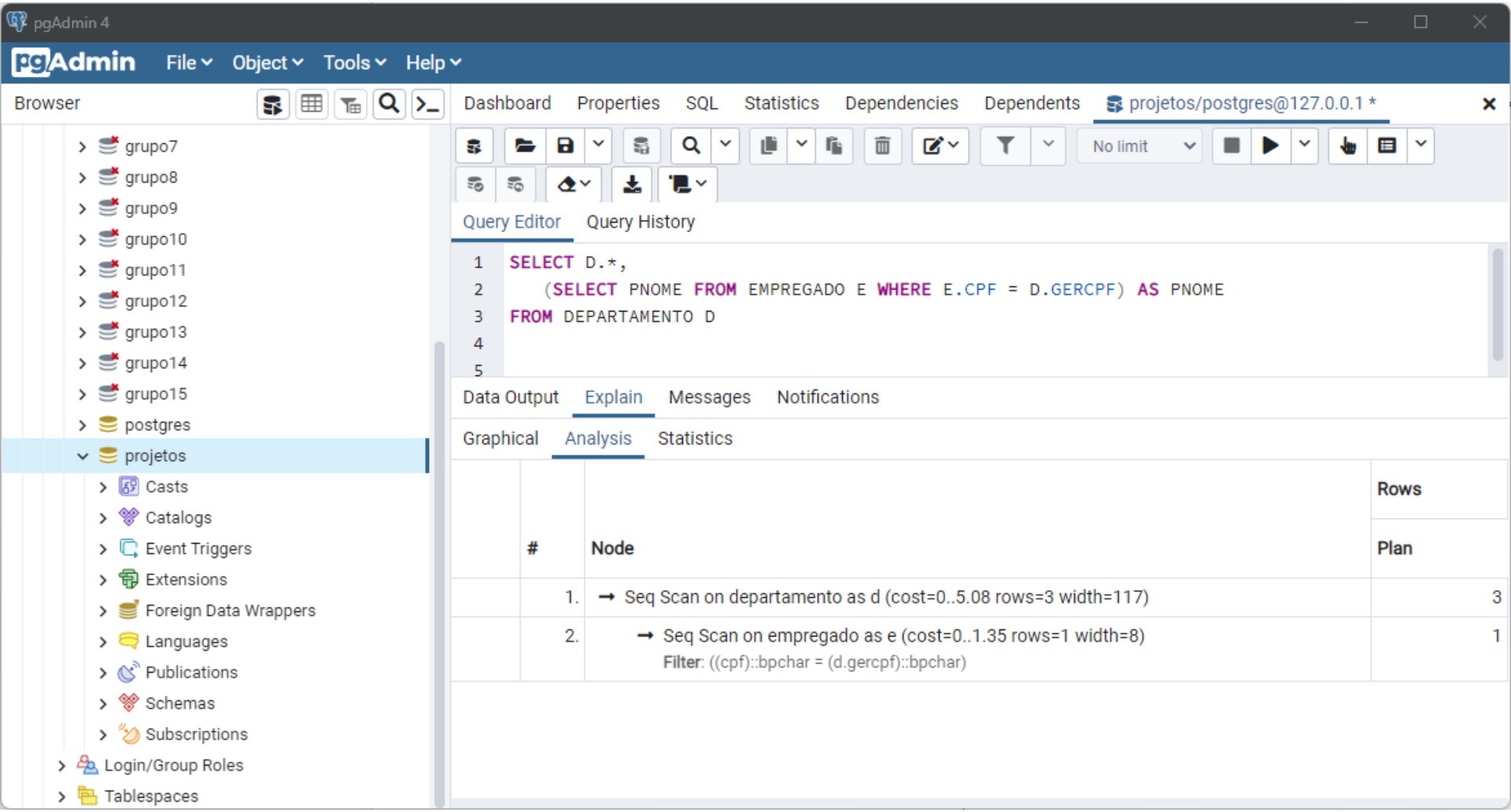
SELECT * FROM EMPREGADO E WHERE E.CPF = '999887777'

Consulta por valor (empregado a partir do primeiro nome)



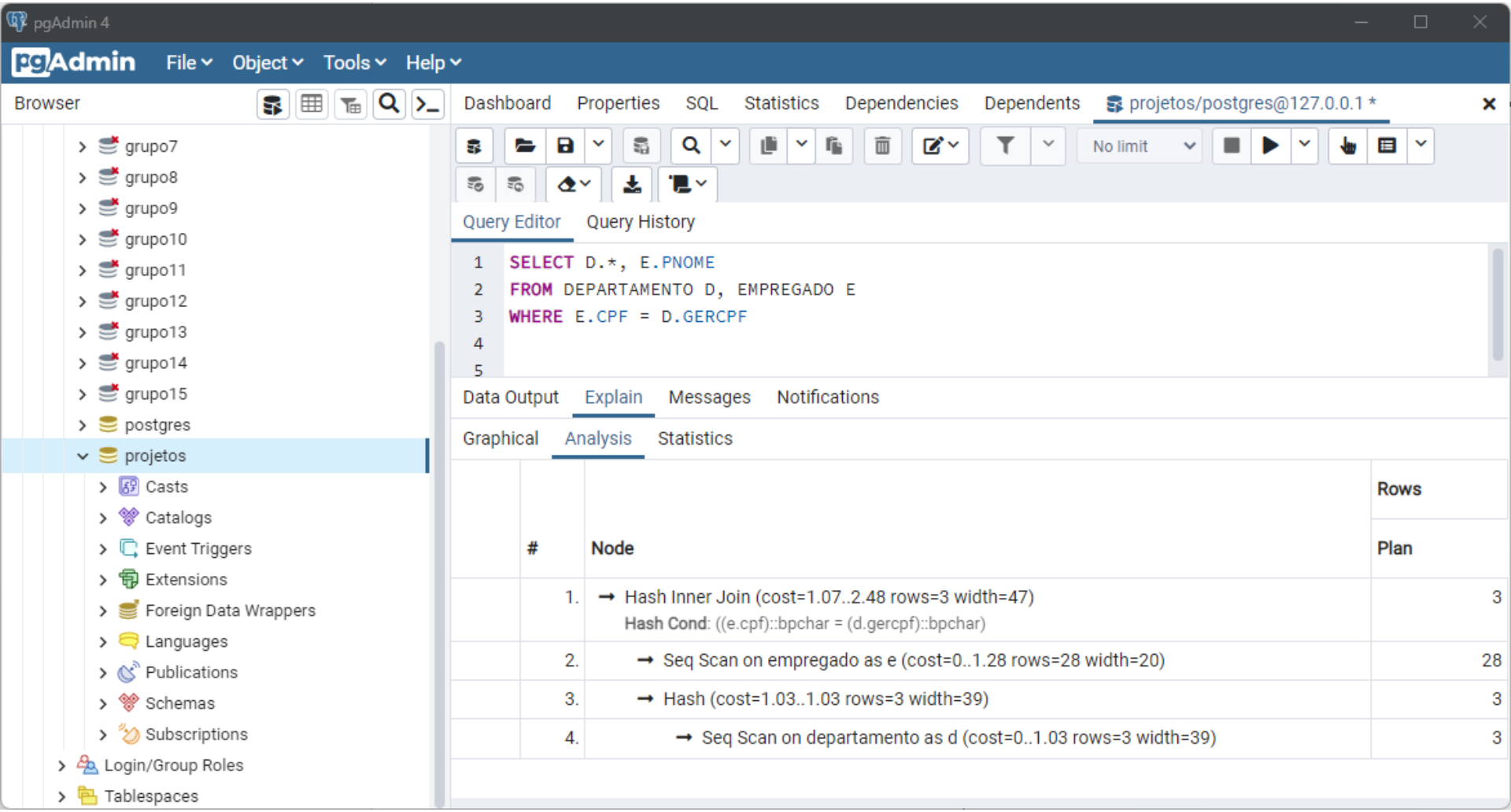
SELECT * FROM EMPREGADO E WHERE E.PNOME = 'Alicia'

Sub-consulta na projeção (lista de departamentos com seus respectivos gerentes)




```
SELECT D.*,
   (SELECT PNOME FROM EMPREGADO E WHERE E.CPF = D.GERCPF) AS PNOME
FROM DEPARTAMENTO D
```

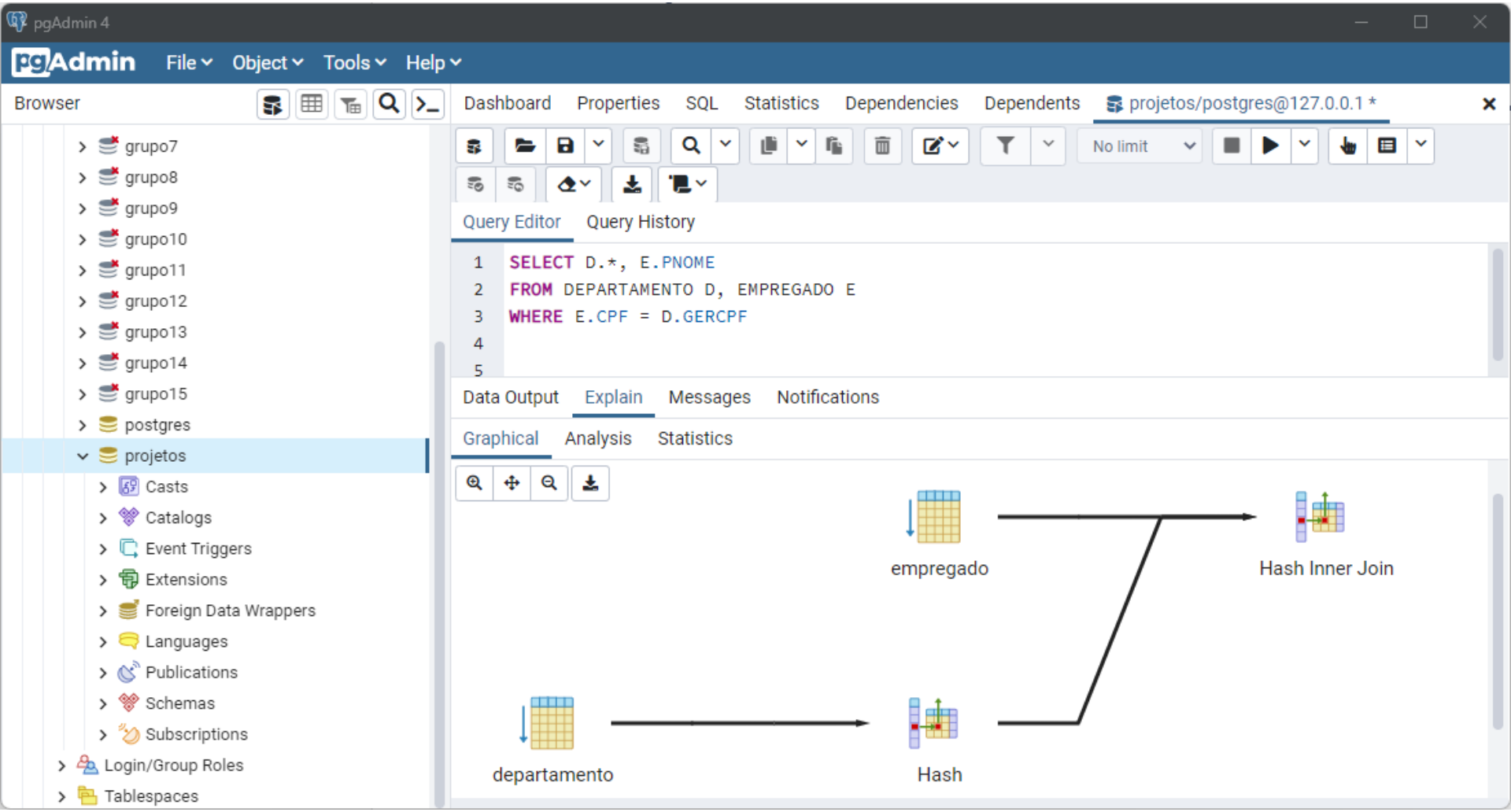
Consulta equivalente em junção
(lista de departamentos com seus respectivos gerentes)



#	Node	Rows Plan
1.	→ Hash Inner Join (cost=1.07..2.48 rows=3 width=47) **Hash Cond**: ((e.cpf)::bpchar = (d.gercpf)::bpchar)	3
2.	→ Seq Scan on empregado as e (cost=0..1.28 rows=28 width=20)	28
3.	→ Hash (cost=1.03..1.03 rows=3 width=39)	3
4.	→ Seq Scan on departamento as d (cost=0..1.03 rows=3 width=39)	3

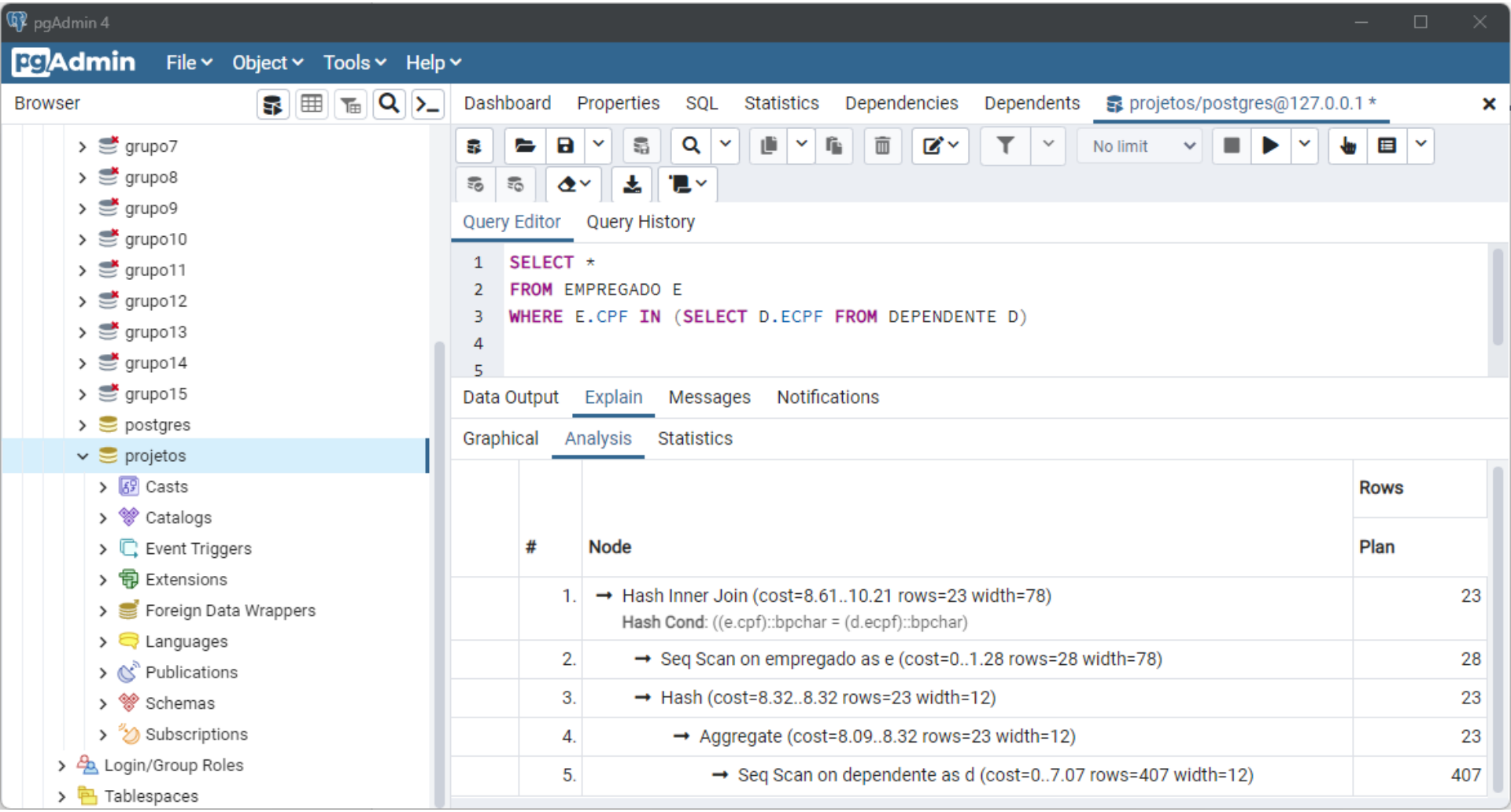
```
SELECT D.*, E.PNOME
FROM DEPARTAMENTO D, EMPREGADO E
WHERE E.CPF = D.GERCPF
```

Árvore de consulta equivalente em junção (lista de departamentos com seus respectivos gerentes)



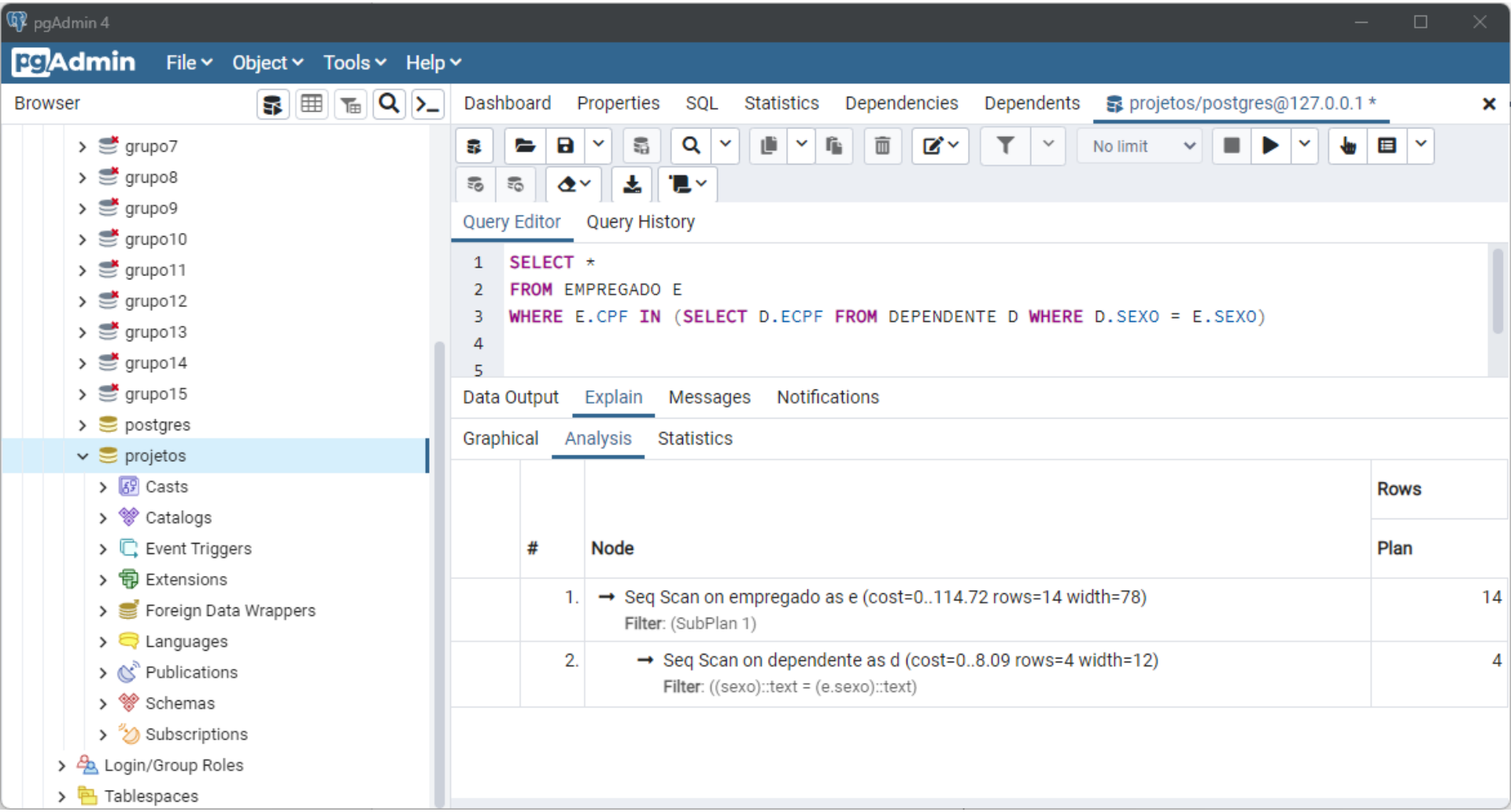

```
SELECT D.*, E.PNOME
FROM DEPARTAMENTO D, EMPREGADO E
WHERE E.CPF = D.GERCPF
```

Subconsulta não correlacionada com in (empregados com dependentes)



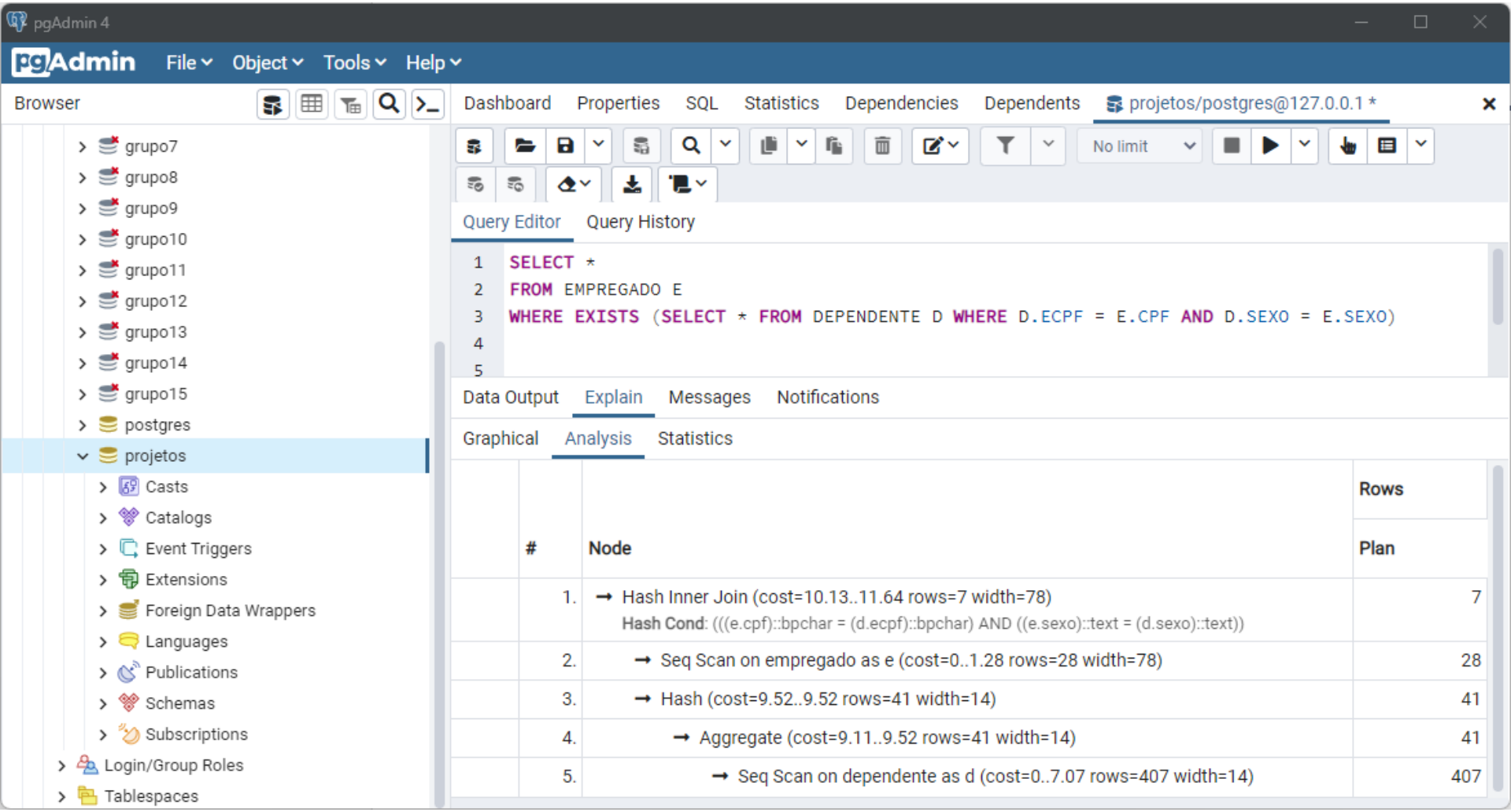

```
SELECT *
FROM EMPREGADO E
WHERE E.CPF IN (SELECT D.ECPF FROM DEPENDENTE D)
```

Subconsulta correlacionada com in (empregados com dependentes do mesmo sexo)




```
SELECT *
FROM EMPREGADO E
WHERE E.CPF IN (SELECT D.ECPF FROM DEPENDENTE D WHERE D.SEXO = E.SEXO)
```

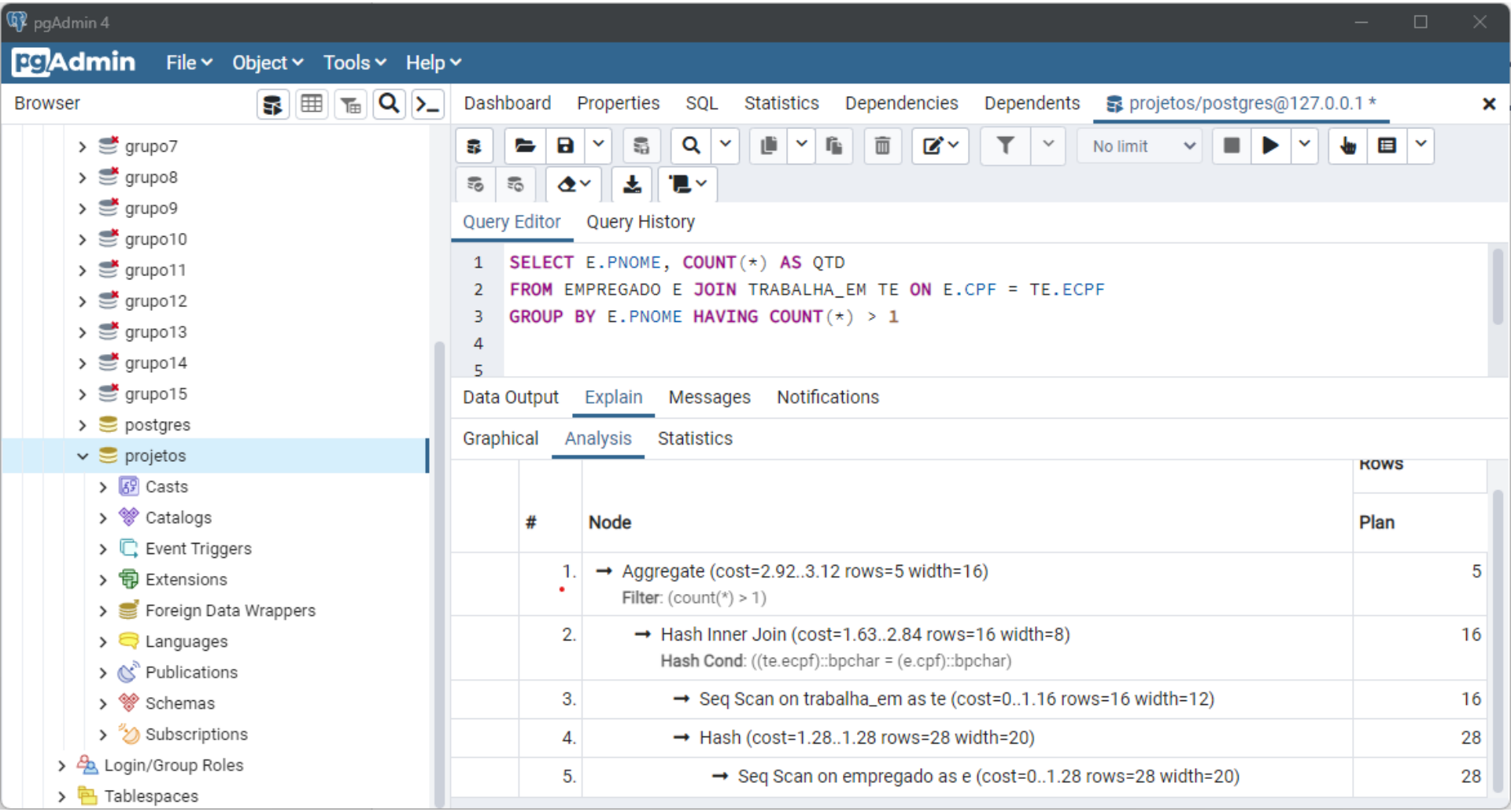
Subconsulta correlacionada com exists (empregados com dependentes do mesmo sexo)



#	Node	Rows Plan
1.	→ Hash Inner Join (cost=10.13..11.64 rows=7 width=78) **Hash Cond**: (((e.cpf)::bpchar = (d.ecpf)::bpchar) AND ((e.sexo)::text = (d.sexo)::text))	7
2.	→ Seq Scan on empregado as e (cost=0..1.28 rows=28 width=78)	28
3.	→ Hash (cost=9.52..9.52 rows=41 width=14)	41
4.	→ Aggregate (cost=9.11..9.52 rows=41 width=14)	41
5.	→ Seq Scan on dependente as d (cost=0..7.07 rows=407 width=14)	407

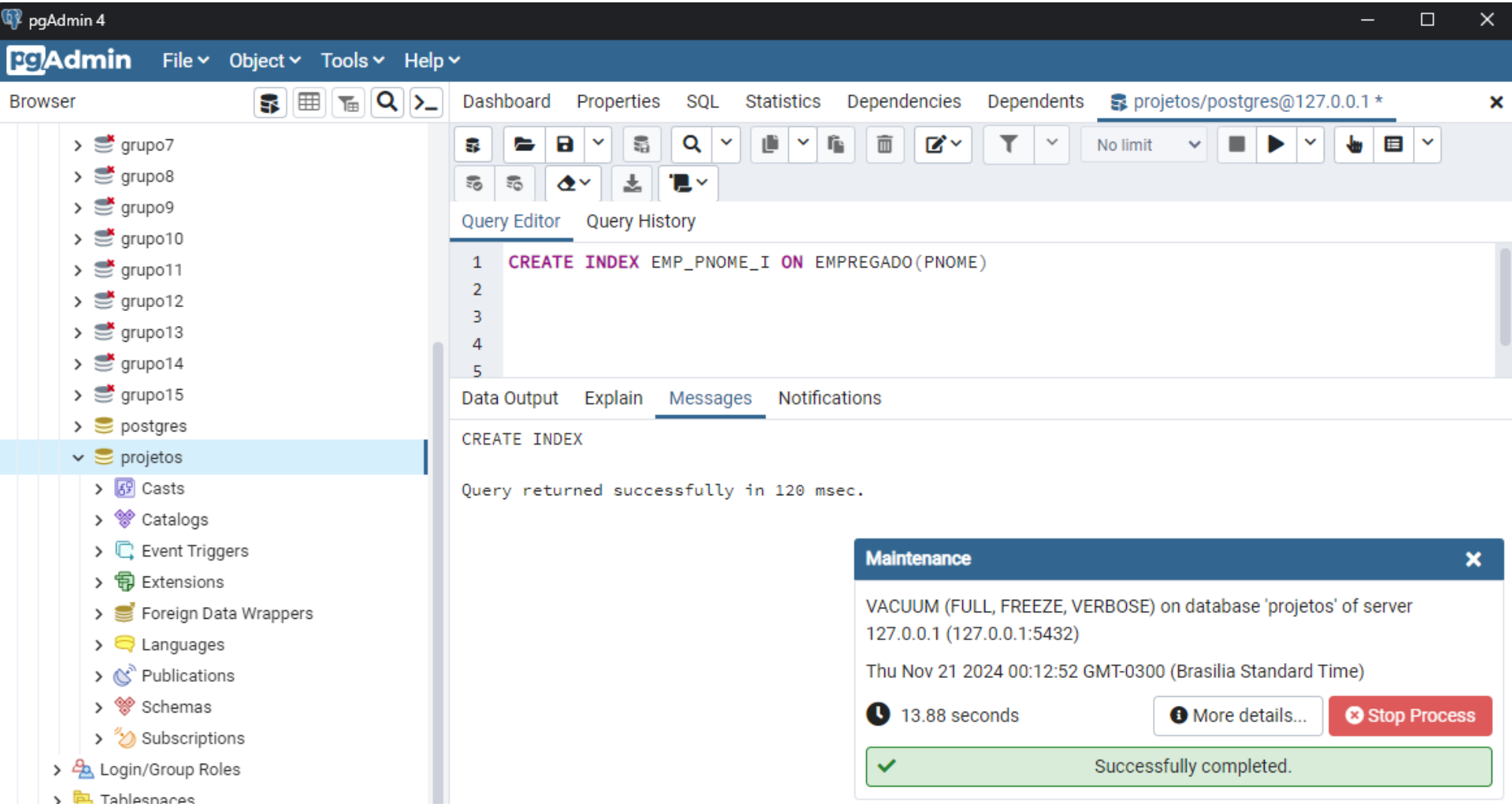
```
SELECT *
FROM EMPREGADO E
WHERE EXISTS (SELECT * FROM DEPENDENTE D WHERE D.ECPF = E.CPF AND D.SEXO = E.SEXO)
```

Consulta com agregação (quantidade de empregados que trabalham em mais de um projeto)



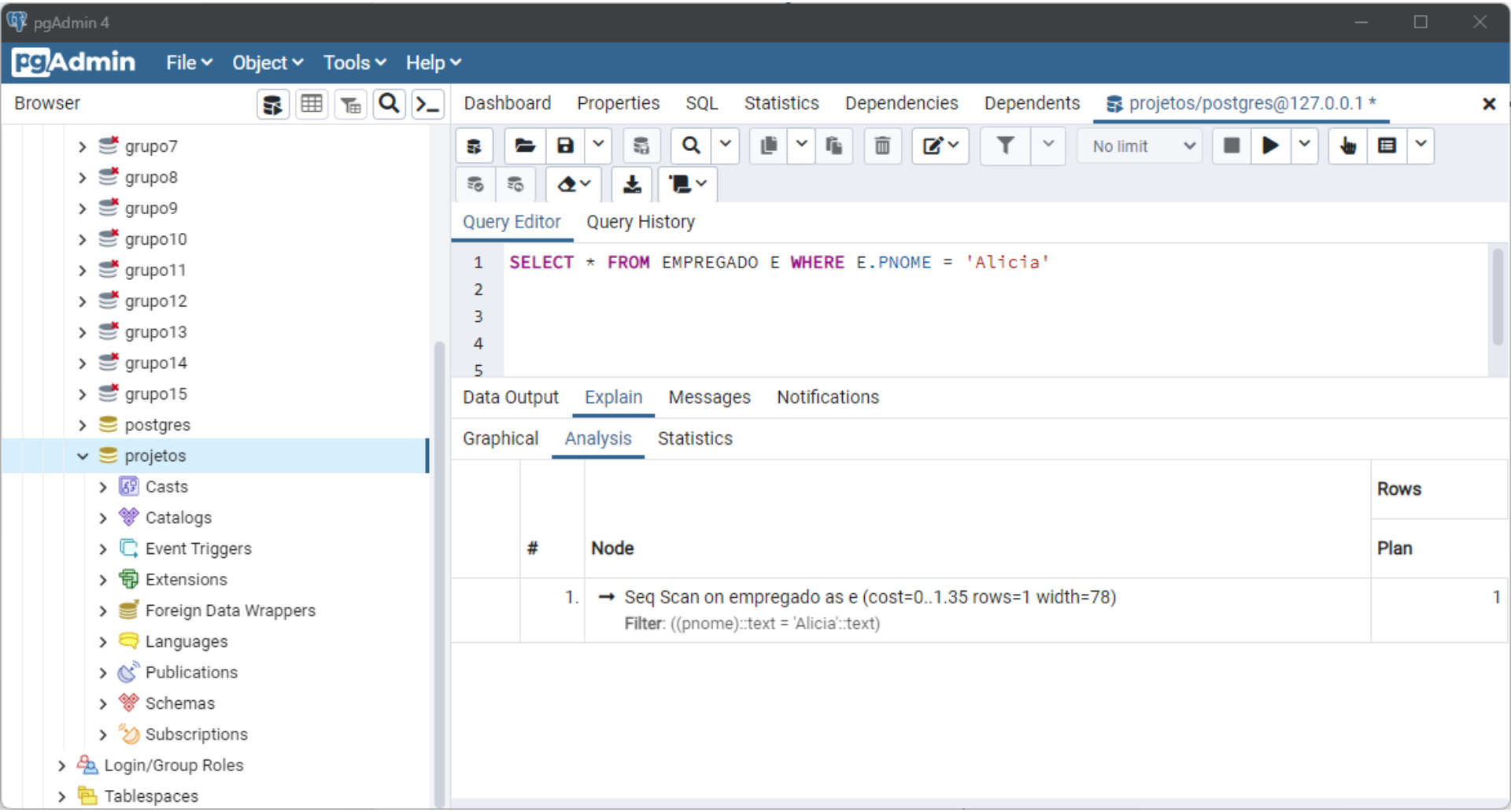

```
SELECT E.PNOME, COUNT(*) AS QTD
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
GROUP BY E.PNOME HAVING COUNT(*) > 1
```

Criação de índice



CREATE INDEX EMP_PNOME_I ON EMPREGADO(PNOME)

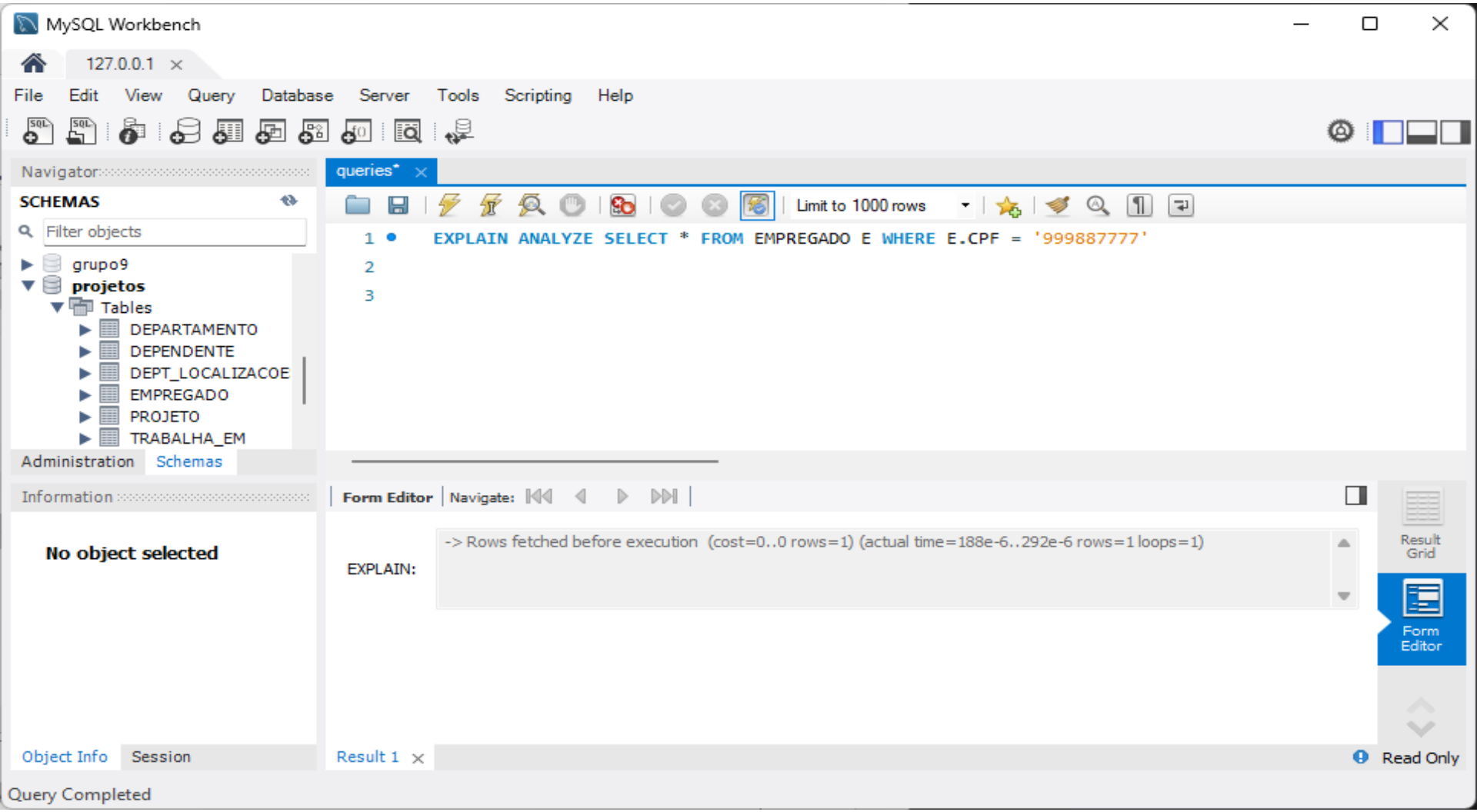
Índice nem sempre é usado (busca pelo empregado a partir do primeiro nome)



SELECT * FROM EMPREGADO E WHERE E.PNOME = 'Alicia'

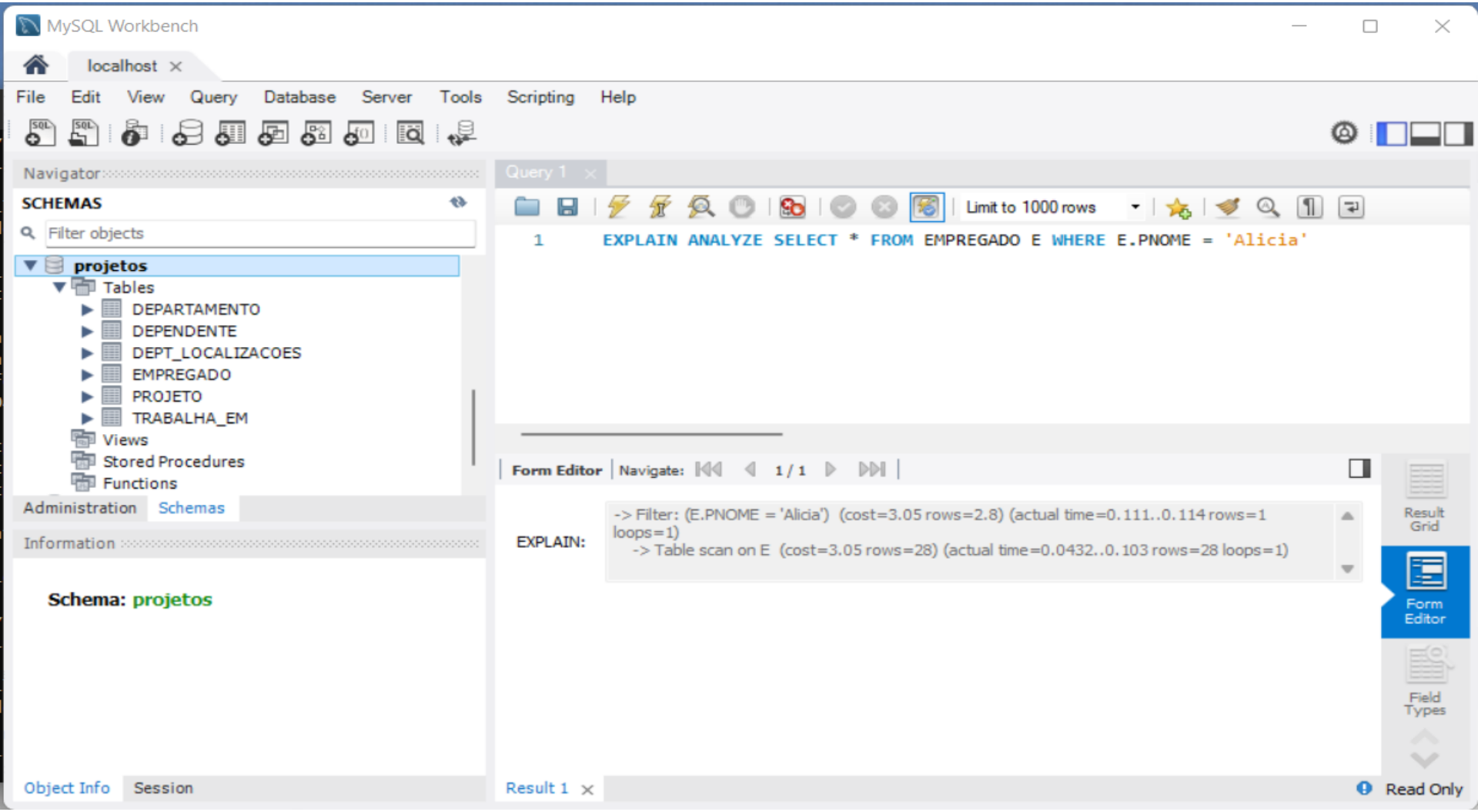
MySQL

Consultas pela chave primária (empregado a partir do cpf)



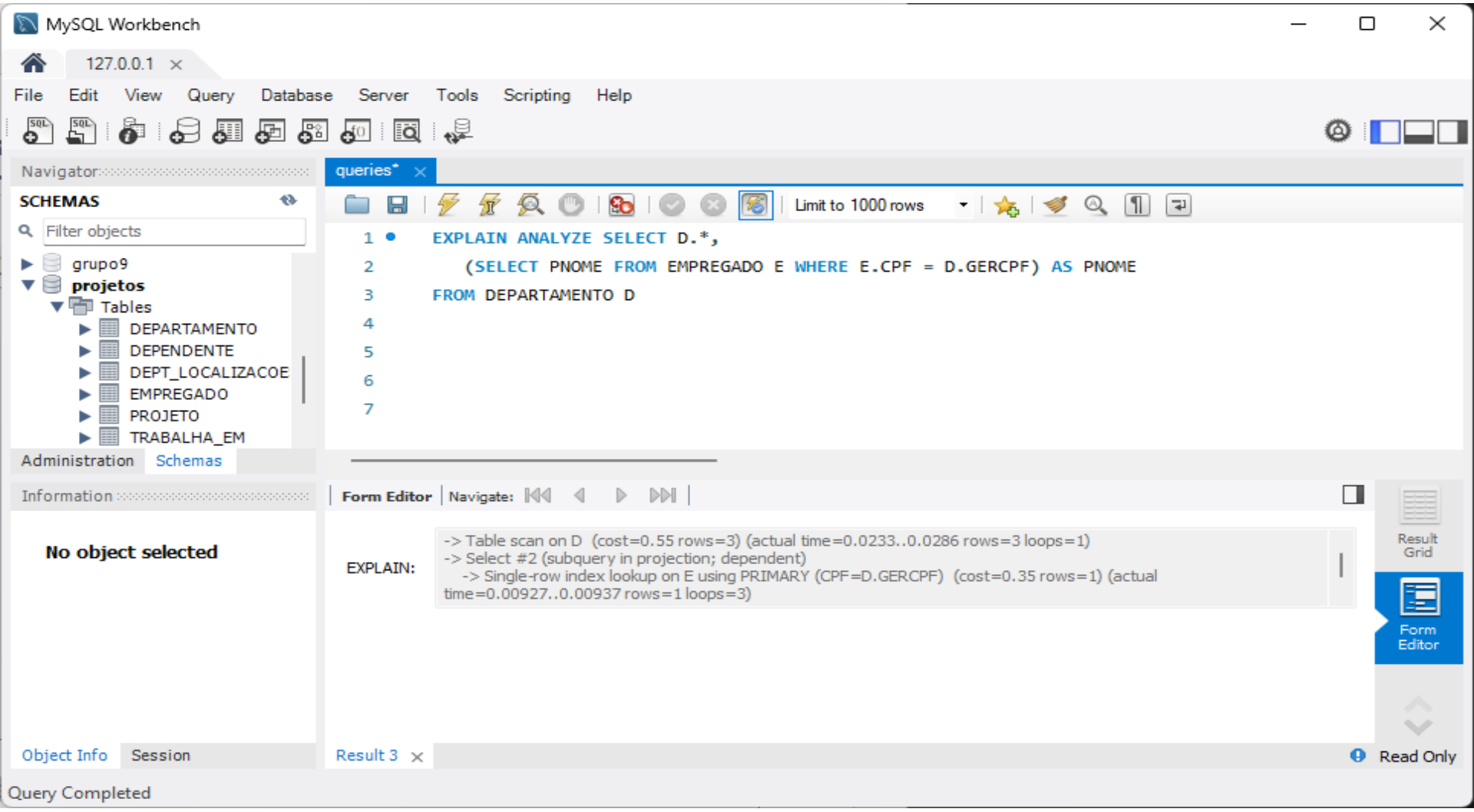
SELECT * FROM EMPREGADO E WHERE E.CPF = '999887777'

Consulta por valor (empregado a partir do primeiro nome)



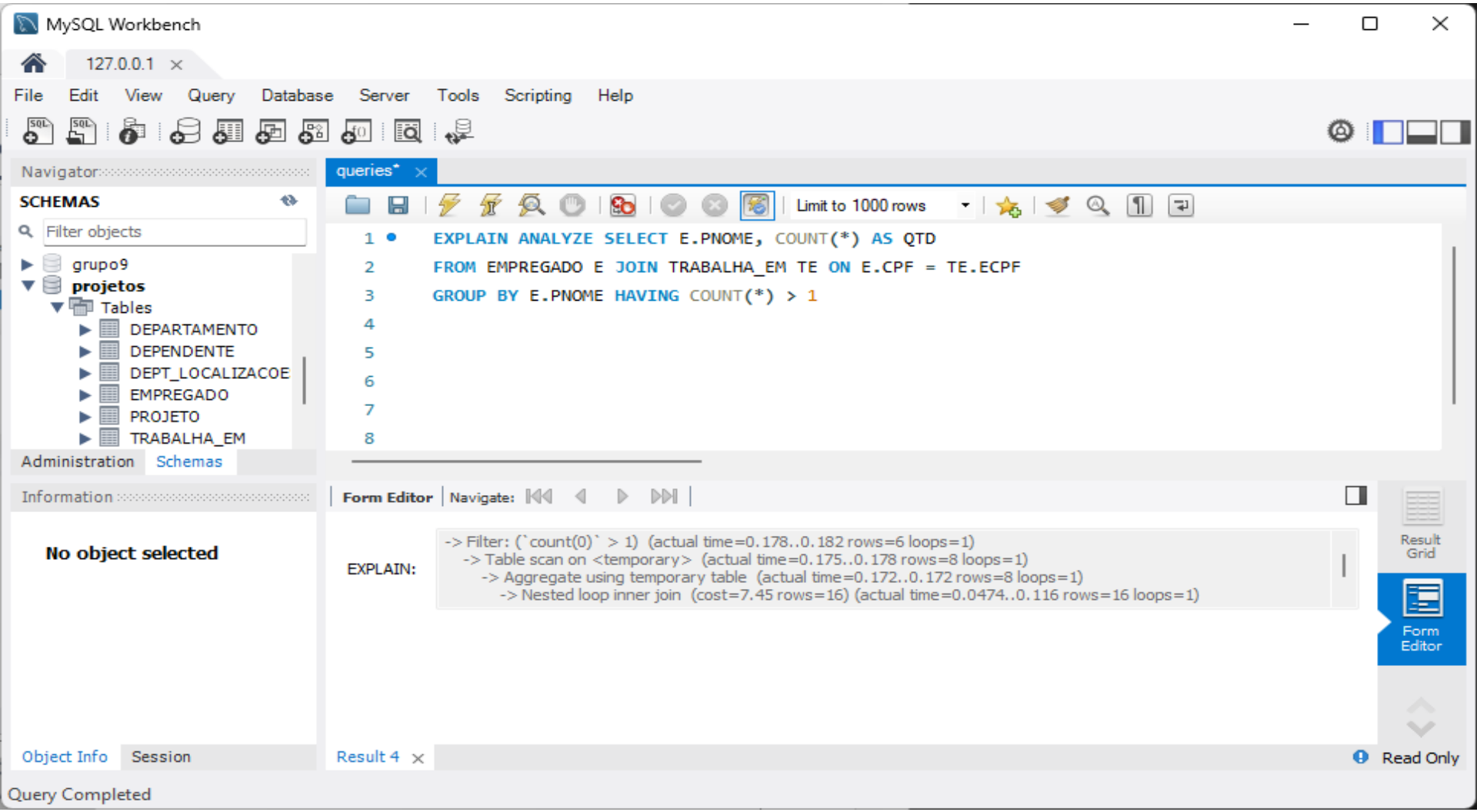
SELECT * FROM EMPREGADO E WHERE E.PNOME = 'Alicia'

Sub-consulta na projeção (lista de departamentos com seus respectivos gerentes)



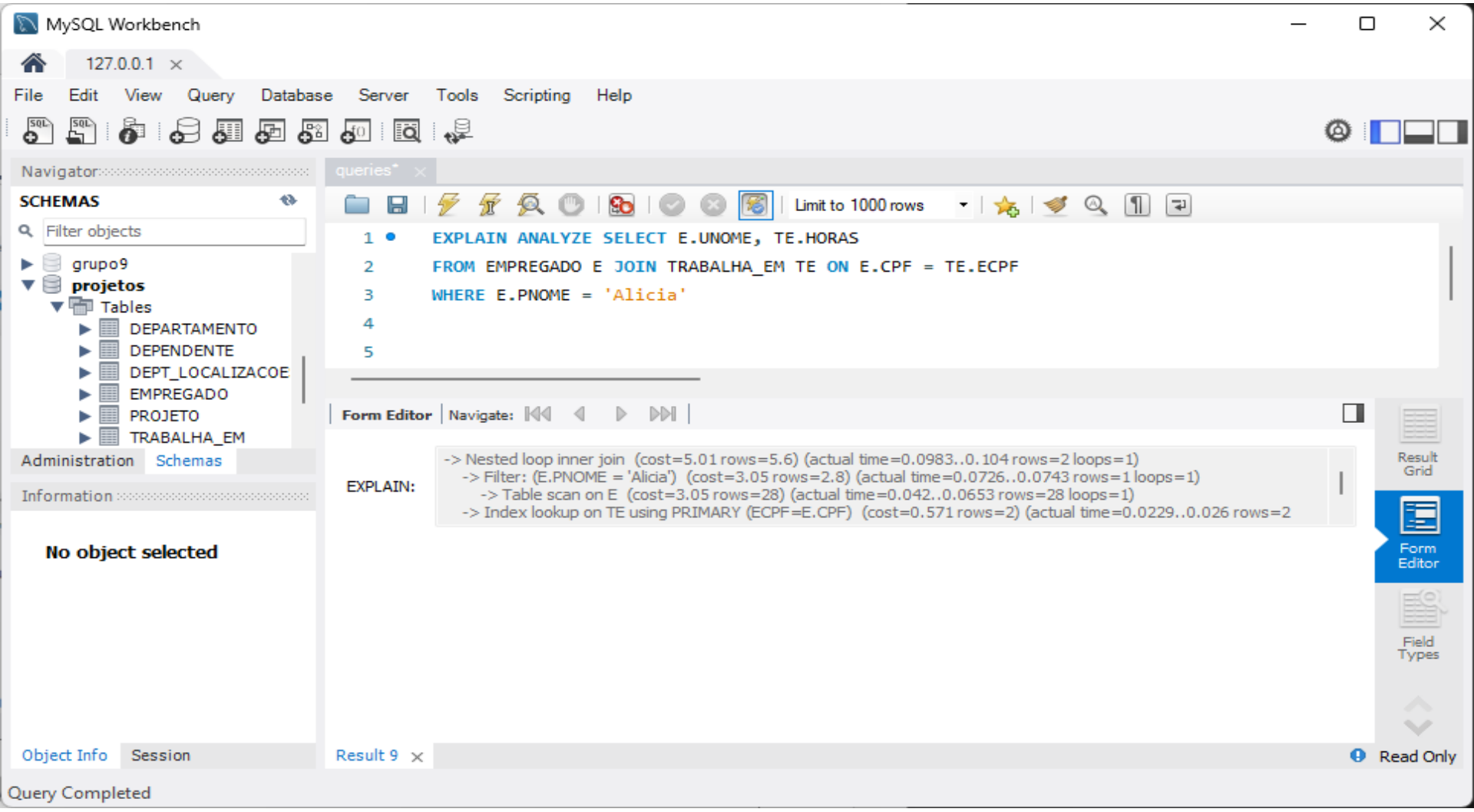

```
SELECT D.*,
  (SELECT PNOME FROM EMPREGADO E WHERE E.CPF = D.GERCPF) AS PNOME
FROM DEPARTAMENTO D
```

Consulta com agregação (quantidade de empregados que trabalham em mais de um projeto)



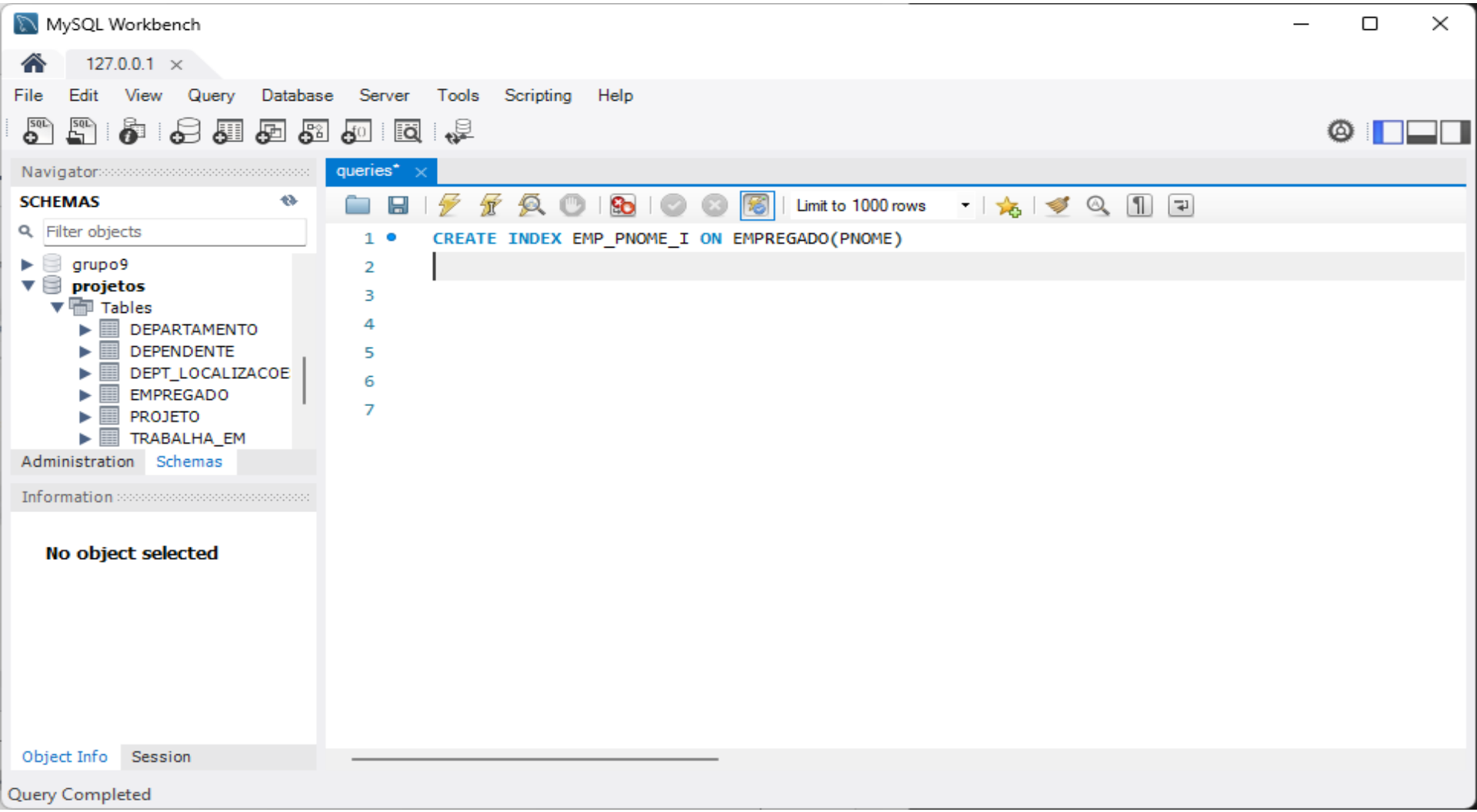

```
SELECT E.PNOME, COUNT(*) AS QTD
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
GROUP BY E.PNOME HAVING COUNT(*) > 1
```

Junção com busca por valor (último nome e horas trabalhadas por projeto pelo primeiro nome)



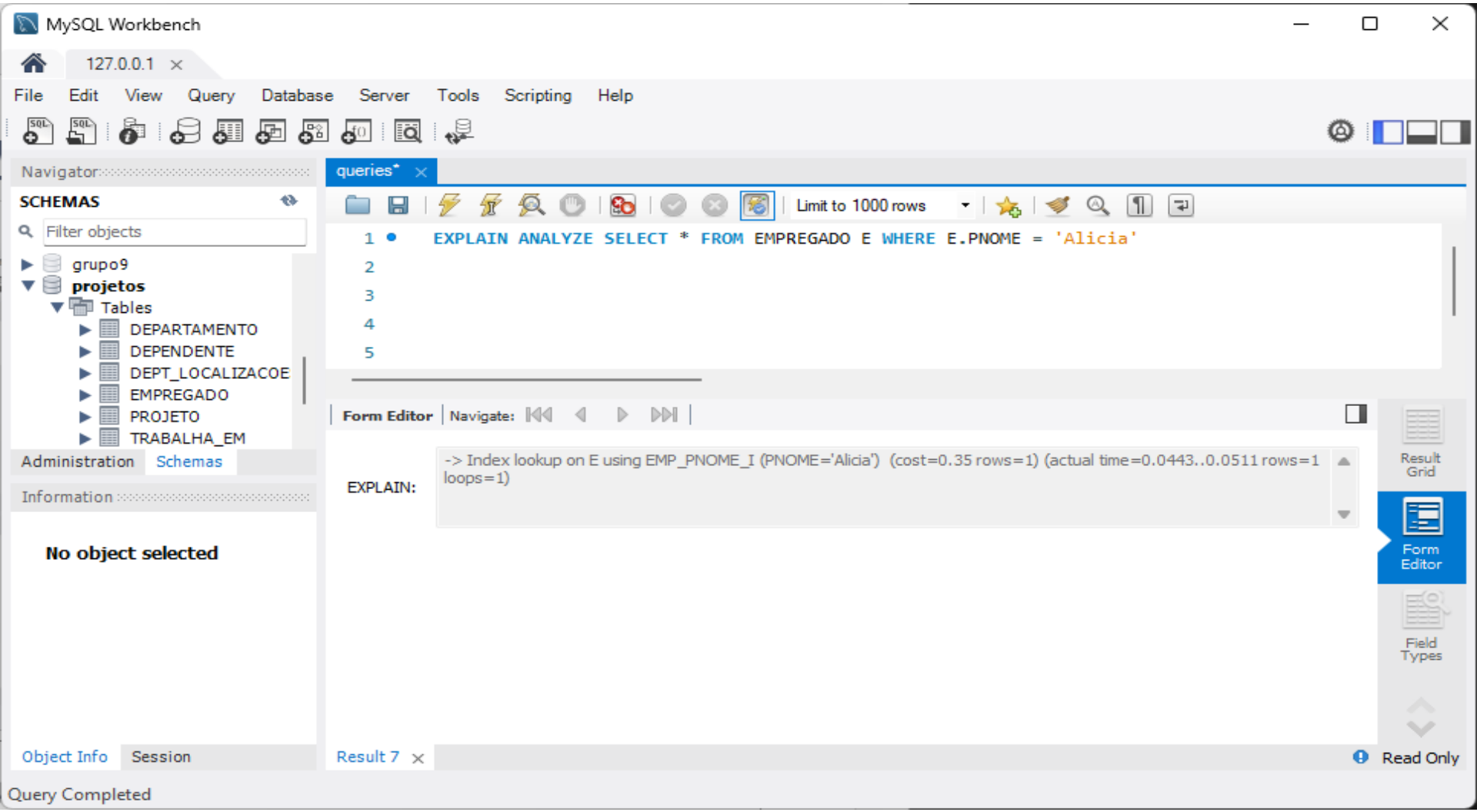

```
SELECT E.UNOME, TE.HORAS
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
WHERE E.PNOME = 'Alicia'
```

Criação de índice



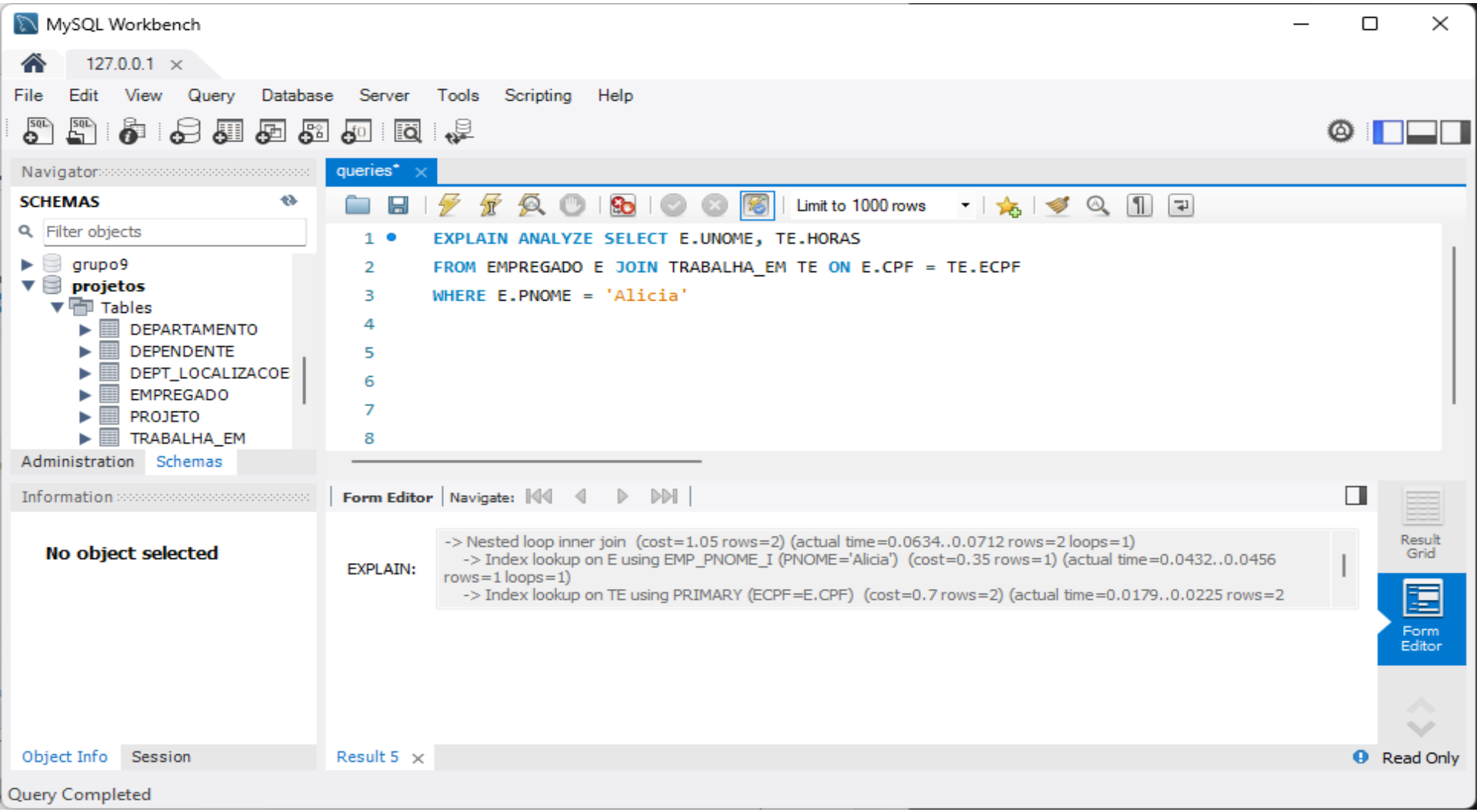
CREATE INDEX EMP_PNOME_I ON EMPREGADO(PNOME)

Uso tradicional do índice (busca pelo empregado a partir do primeiro nome)



SELECT * FROM EMPREGADO E WHERE E.PNOME = 'Alicia'

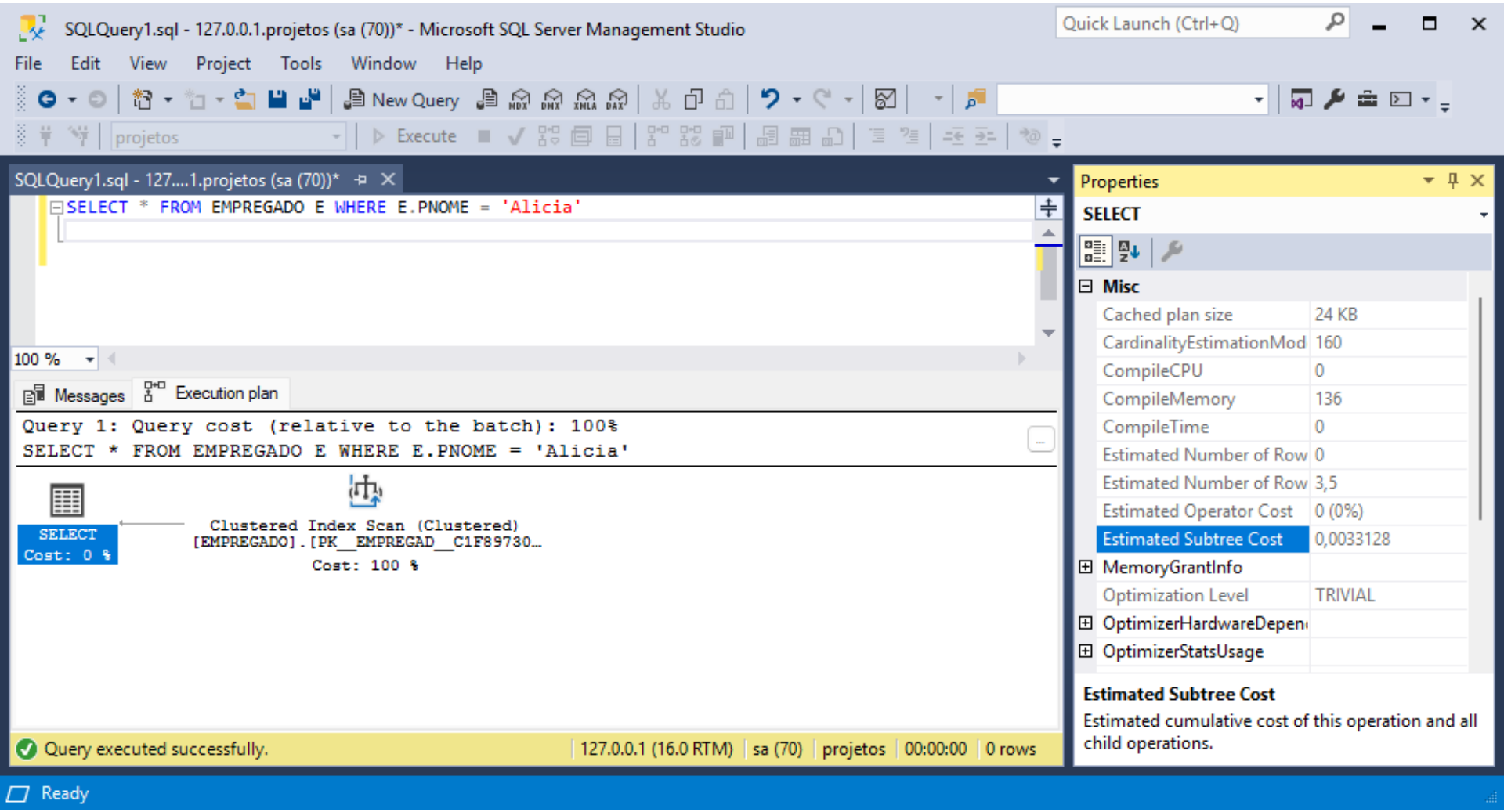
Junção tradicional (índice do nome do empregado foi usado)




```
SELECT E.UNOME, TE.HORAS
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
WHERE E.PNOME = 'Alicia'
```

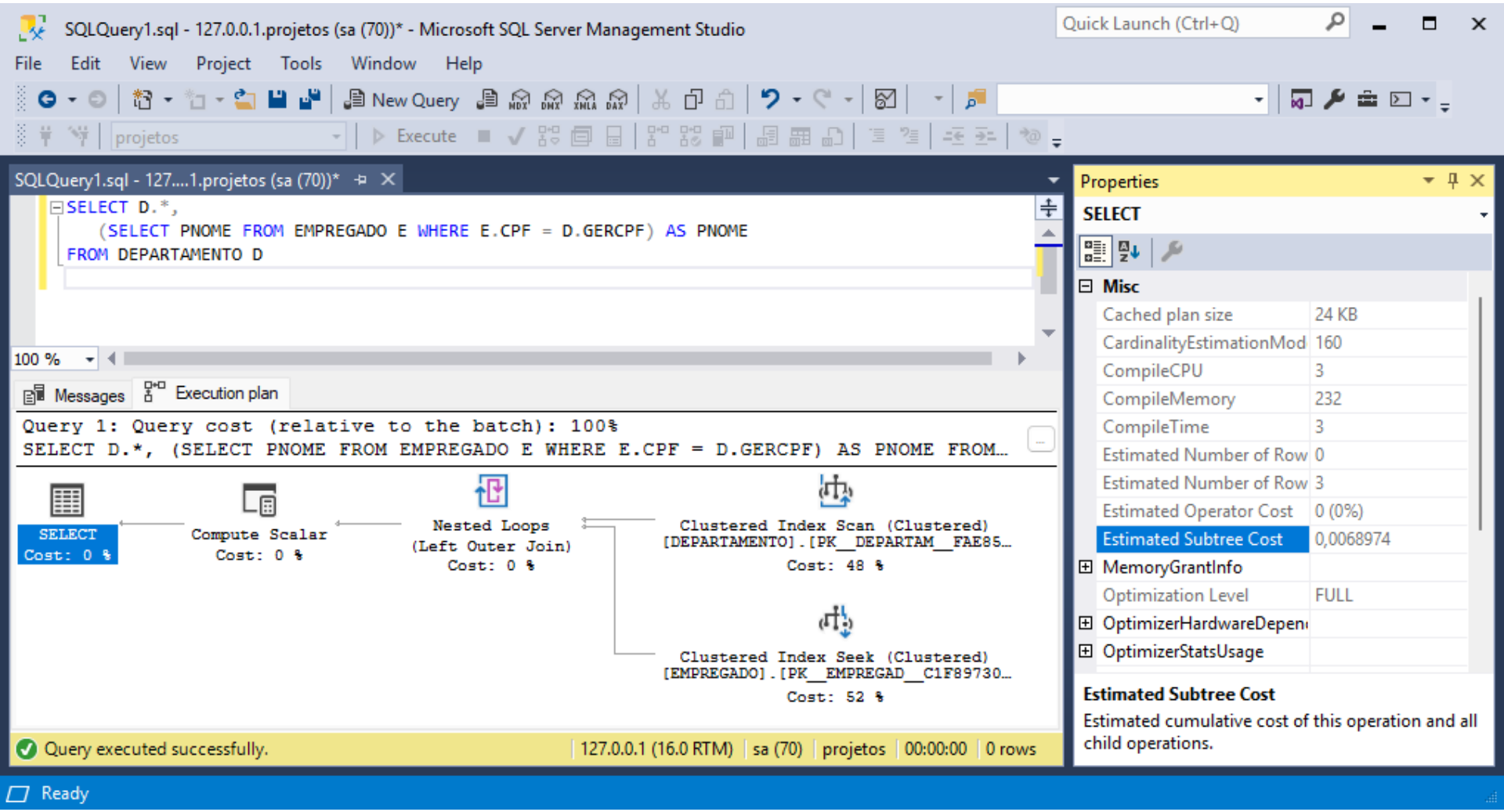
SQL Server

Consulta por valor (empregado a partir do primeiro nome)



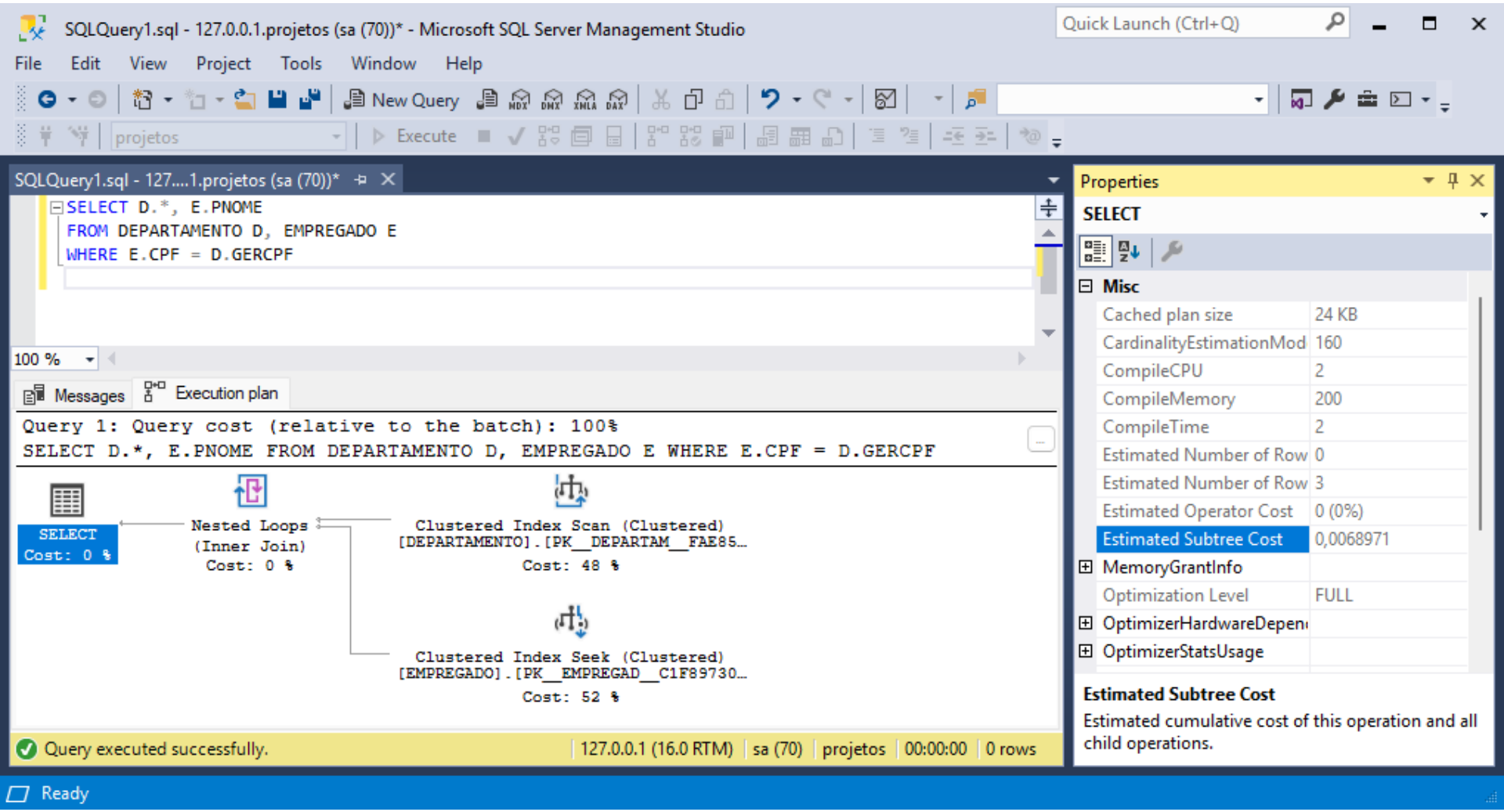
SELECT * FROM EMPREGADO E WHERE E.PNOME = 'Alicia'

Sub-consulta na projeção (lista de departamentos com seus respectivos gerentes)



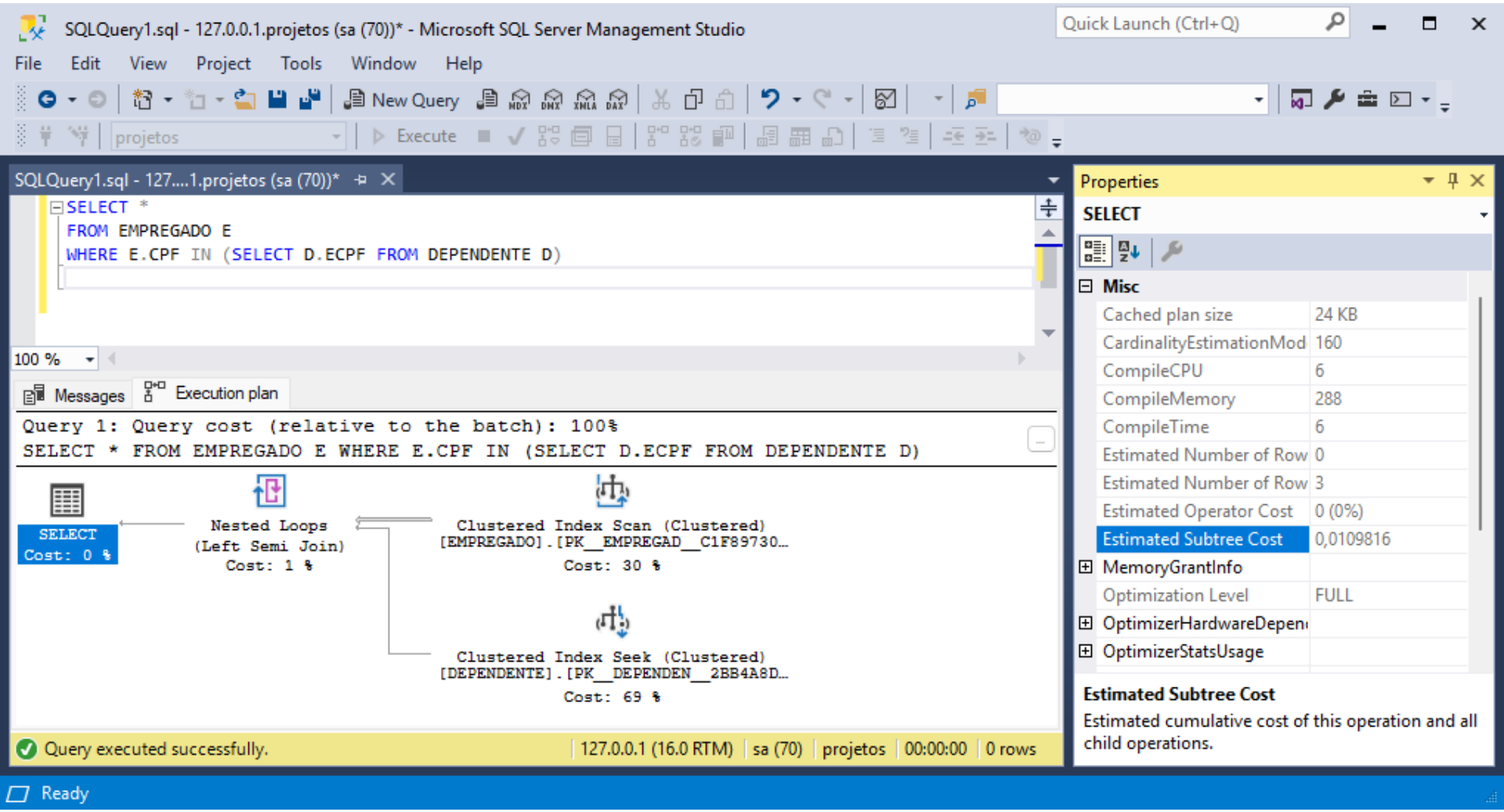

```
SELECT D.*,
   (SELECT PNOME FROM EMPREGADO E WHERE E.CPF = D.GERCPF) AS PNOME
FROM DEPARTAMENTO D
```

Consulta equivalente em junção (lista de departamentos com seus respectivos gerentes)



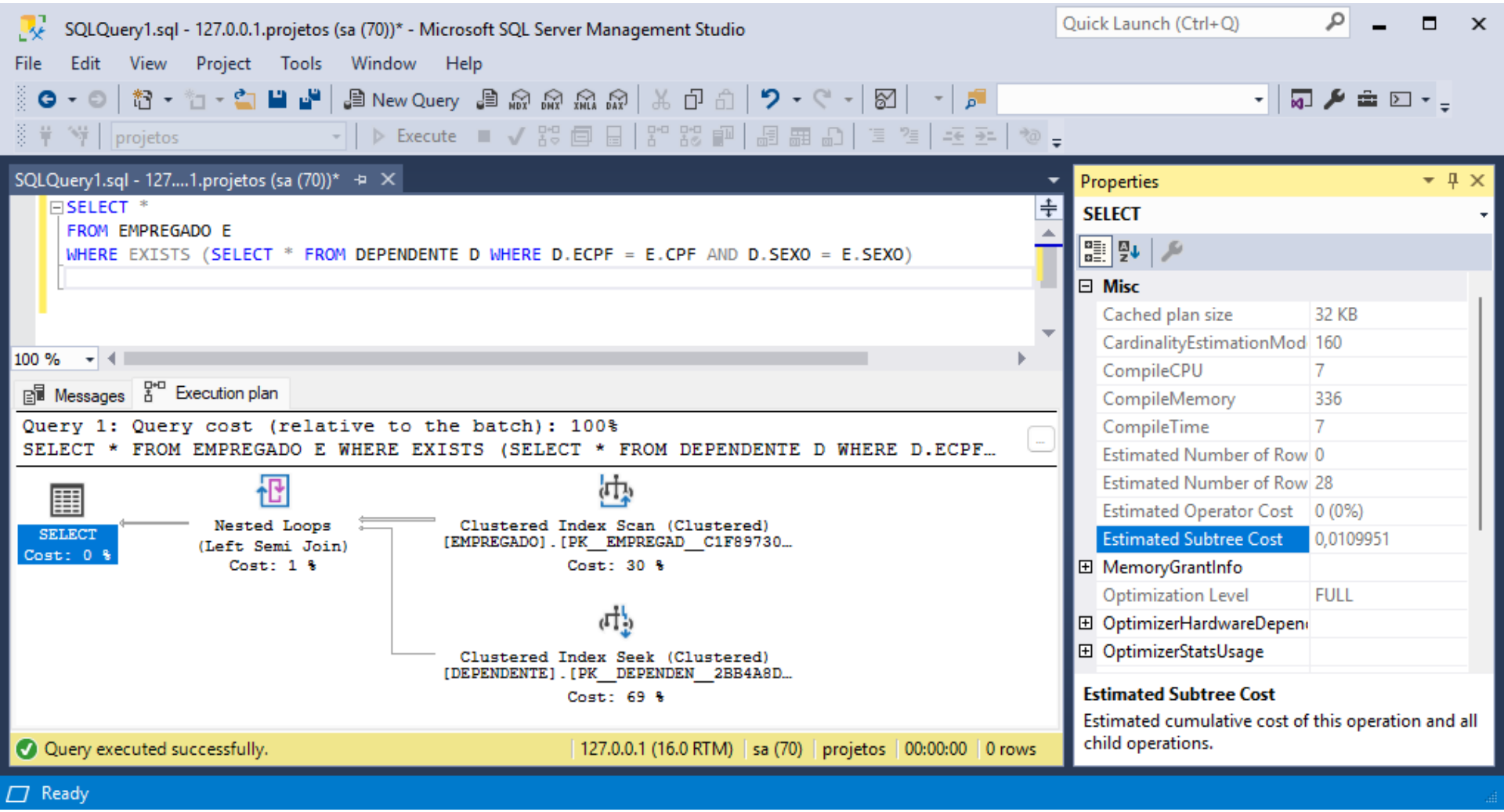

```
SELECT D.*, E.PNOME
FROM DEPARTAMENTO D, EMPREGADO E
WHERE E.CPF = D.GERCPF
```

Subconsulta não correlacionada com in (empregados com dependentes)



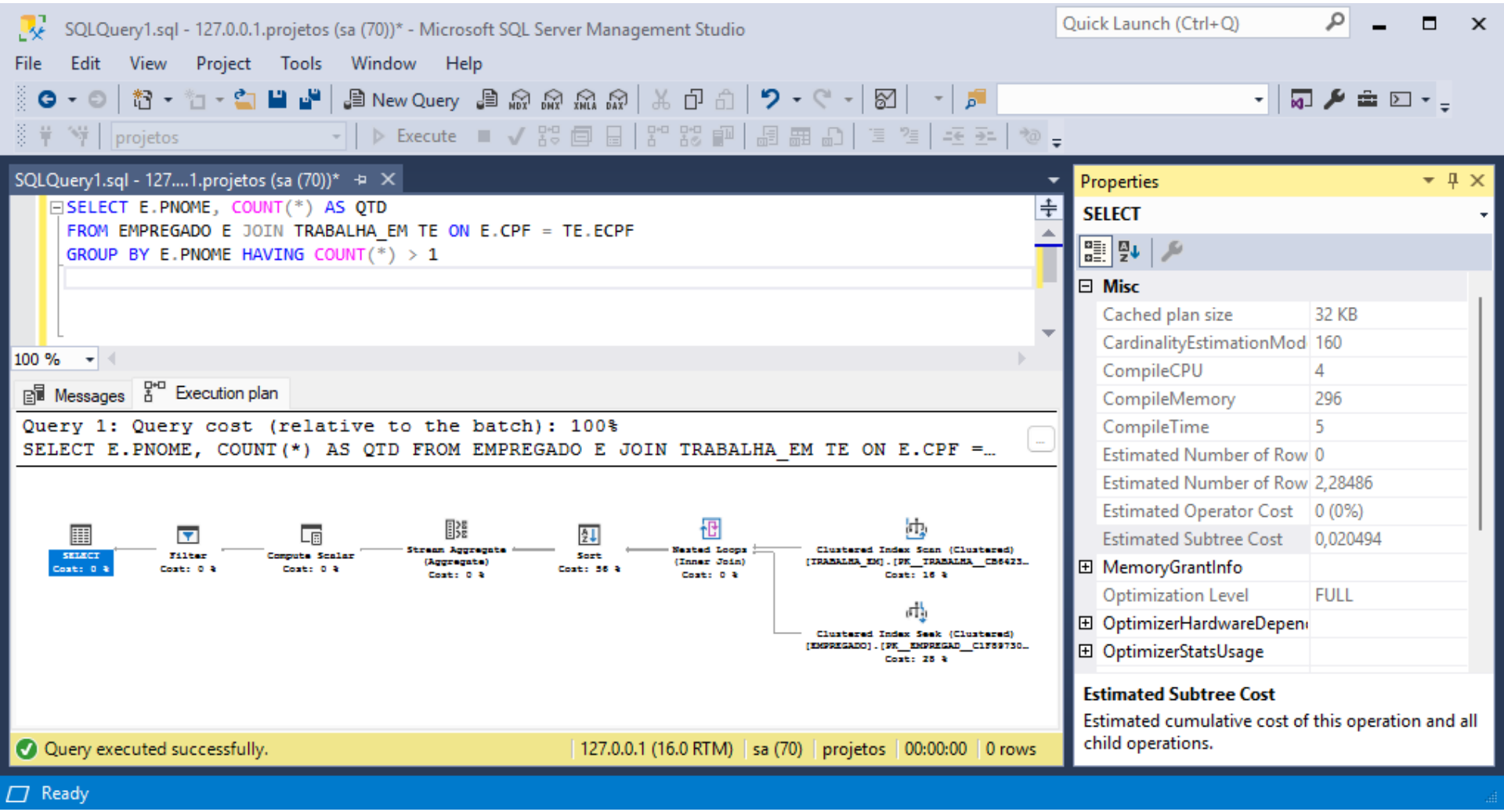

```
SELECT *
FROM EMPREGADO E
WHERE E.CPF IN (SELECT D.ECPF FROM DEPENDENTE D)
```

Subconsulta correlacionada com exists (empregados com dependentes do mesmo sexo)



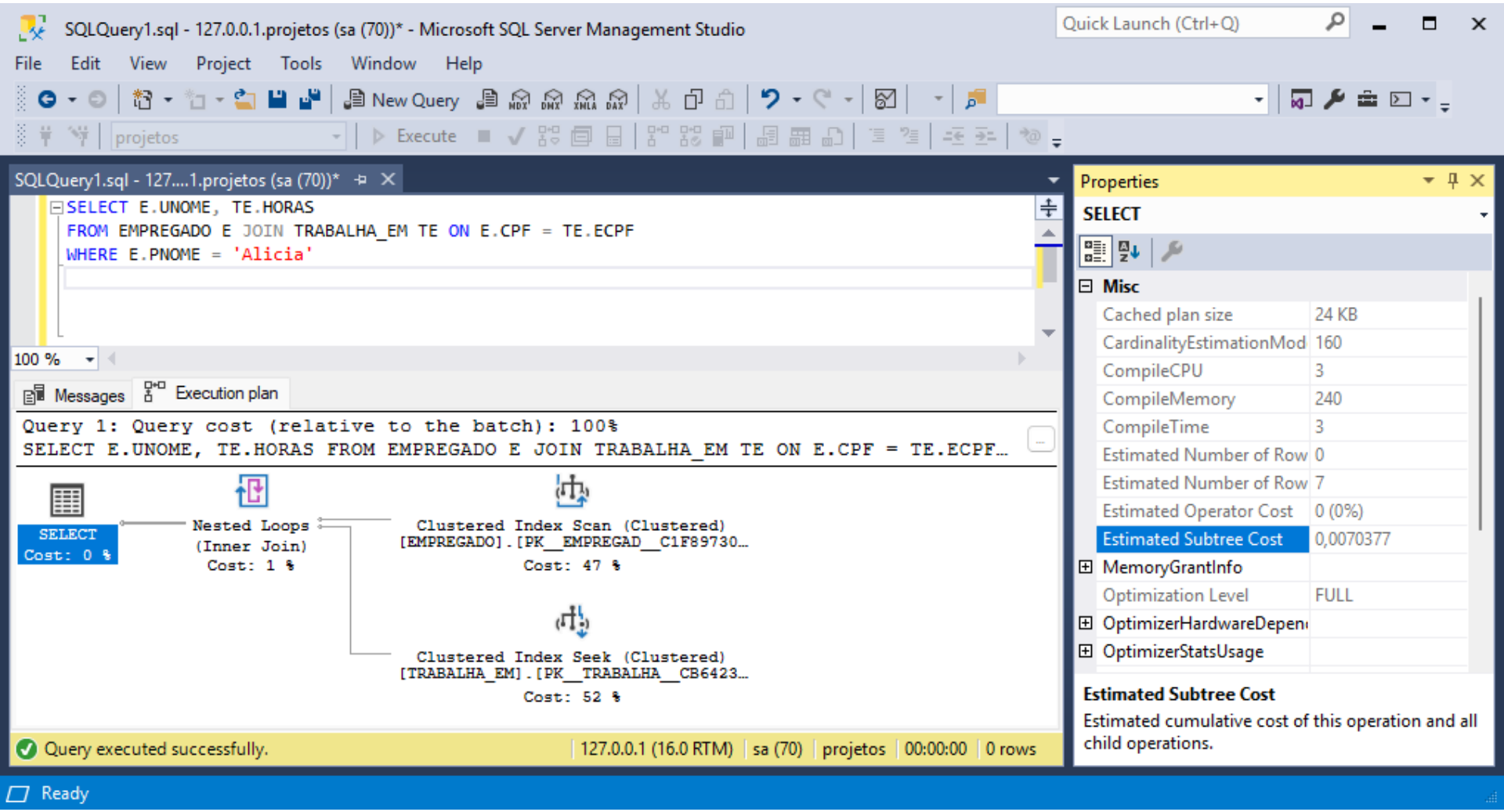

```
SELECT *
FROM EMPREGADO E
WHERE EXISTS (SELECT * FROM DEPENDENTE D WHERE D.ECPF = E.CPF AND D.SEXO = E.SEXO)
```

Consulta com agregação (quantidade de empregados que trabalham em mais de um projeto)




```
SELECT E.PNOME, COUNT(*) AS QTD
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
GROUP BY E.PNOME HAVING COUNT(*) > 1
```

Junção com busca por valor (último nome e horas trabalhadas por projeto pelo primeiro nome)

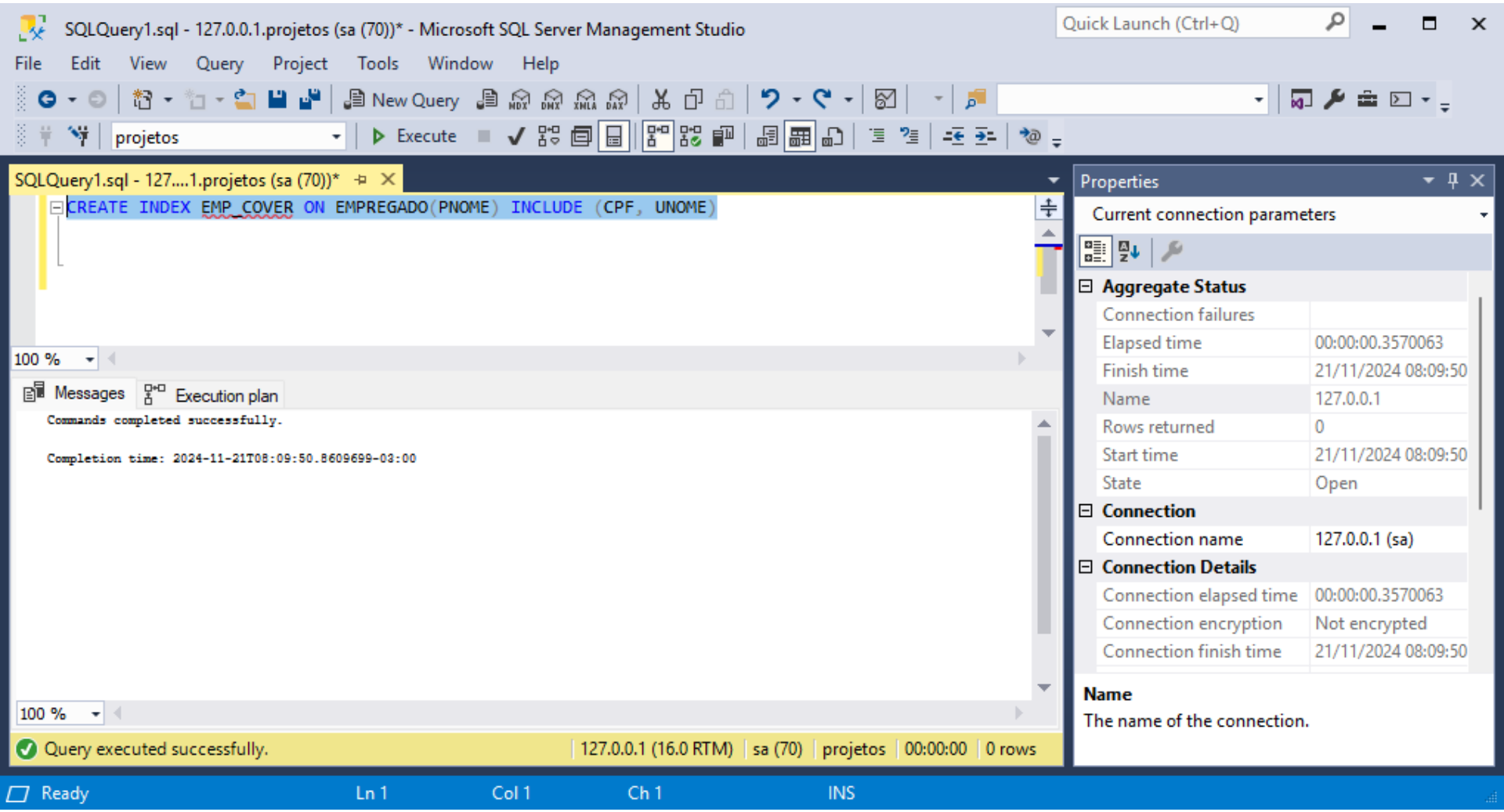



```
SELECT E.UNOME, TE.HORAS
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
WHERE E.PNOME = 'Alicia'
```

Índice de cobertura de consulta

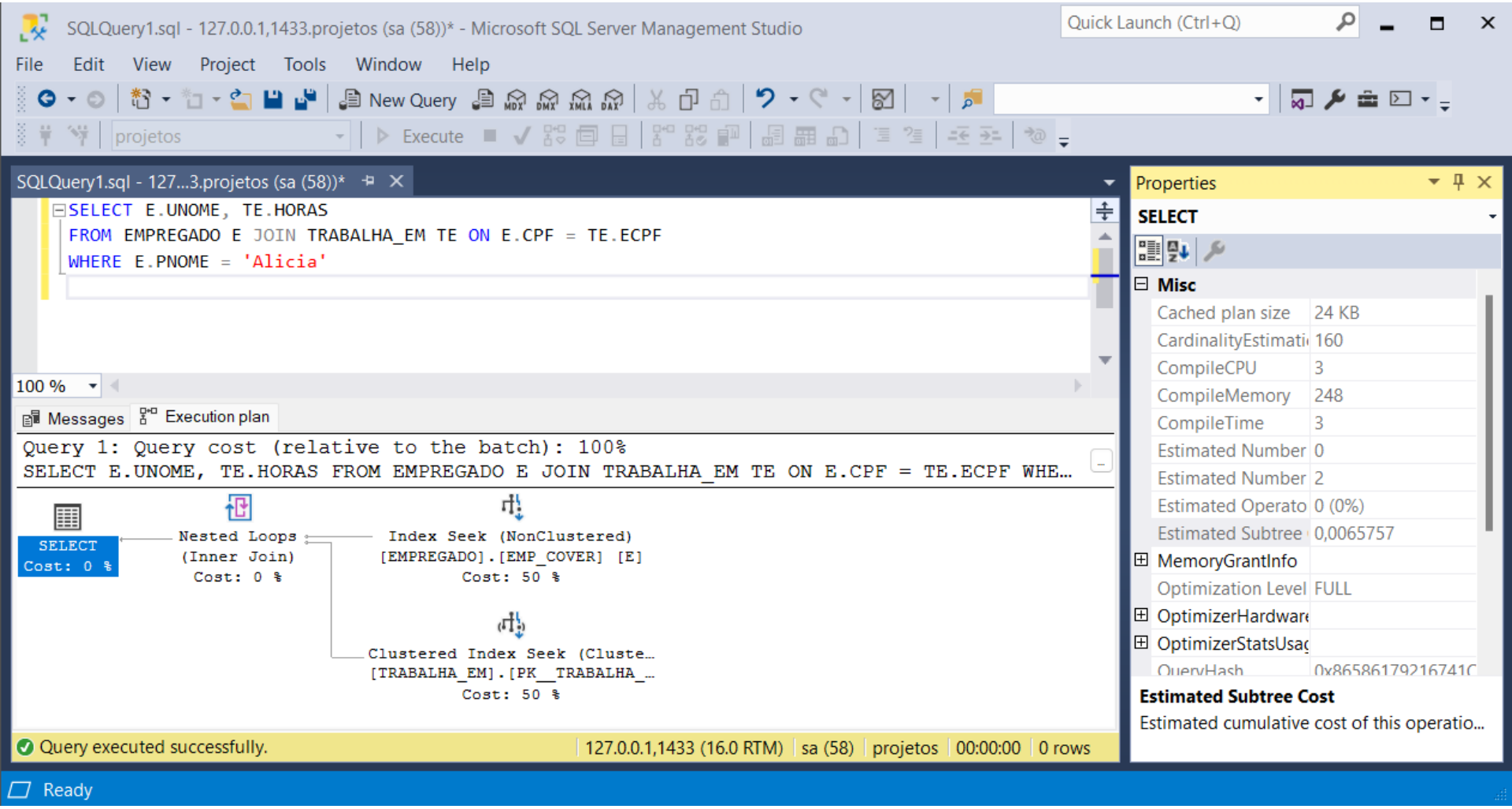
- O índice de cobertura de consulta possibilita processar a consulta sem selecionar a tabela
- Todos os atributos usados na consulta para uma Tabela T estão presentes no índice
- A ordem de criação dos campos no índice é relevante
  - Primeiro os atributos de seleção (mais restritivos primeiro)
  - Segundo os atributos usados na junção
  - Terceiro os atributos usados na projeção

Índice de cobertura de consulta



CREATE INDEX EMP_COVER ON EMPREGADO(PNOME) INCLUDE (CPF, UNOME)

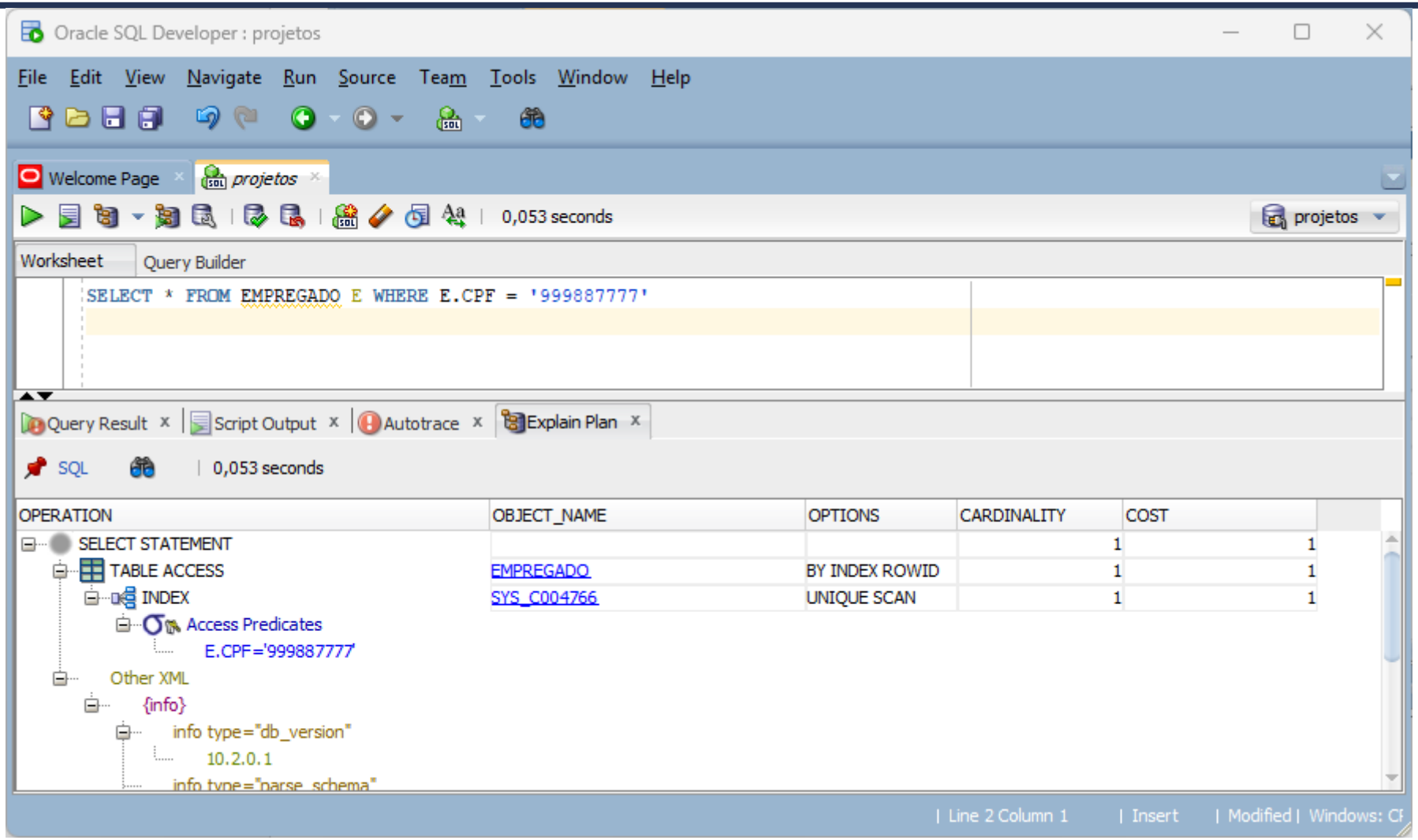
Uso do índice de cobertura de consulta (Consulta de alocação por data de nascimento com índice)




```
SELECT E.UNOME, TE.HORAS
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
WHERE E.PNOME = 'Alicia'
```

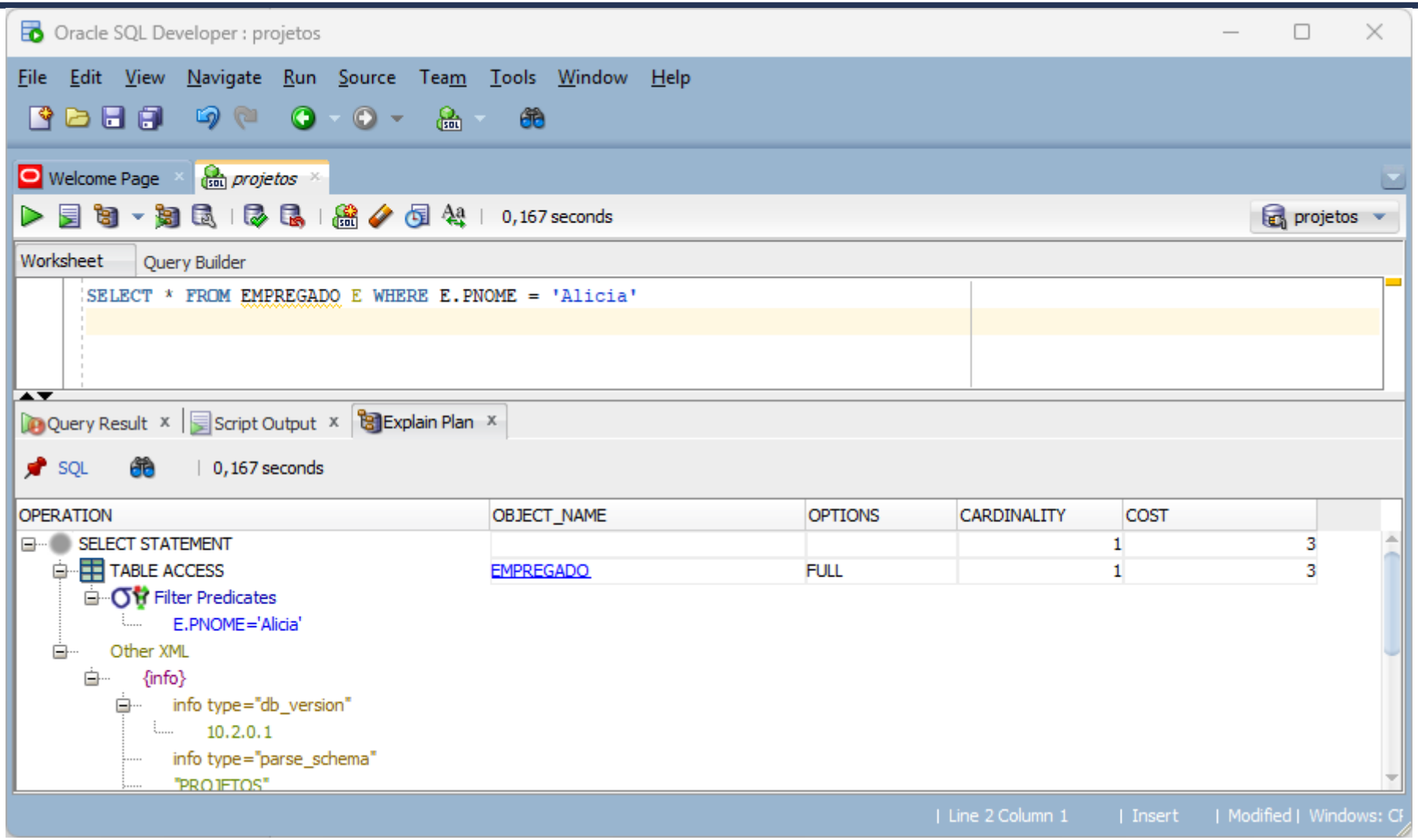
Oracle

Consultas pela chave primária (empregado a partir do cpf)



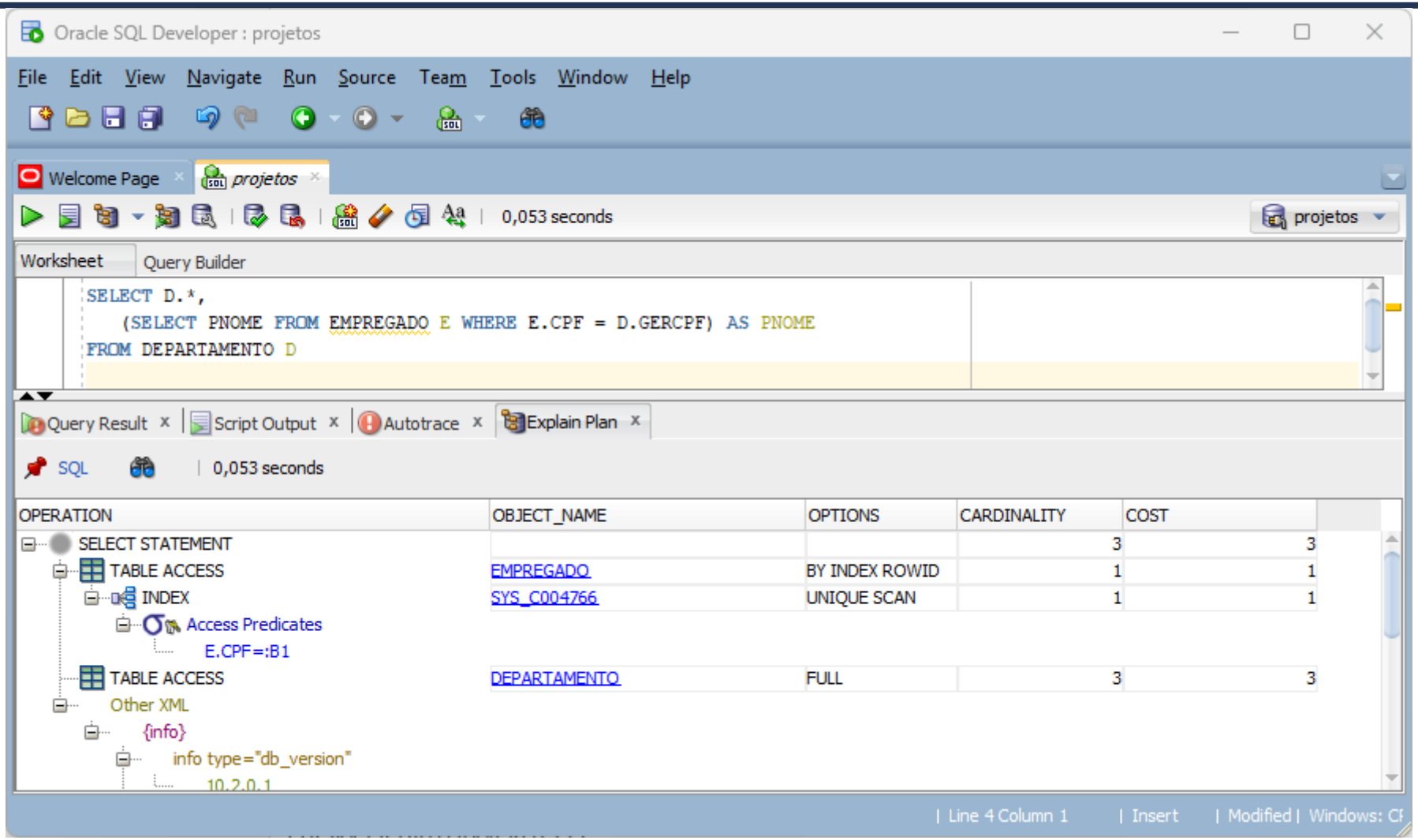
SELECT * FROM EMPREGADO E WHERE E.CPF = '999887777'

Consulta por valor (empregado a partir do primeiro nome)



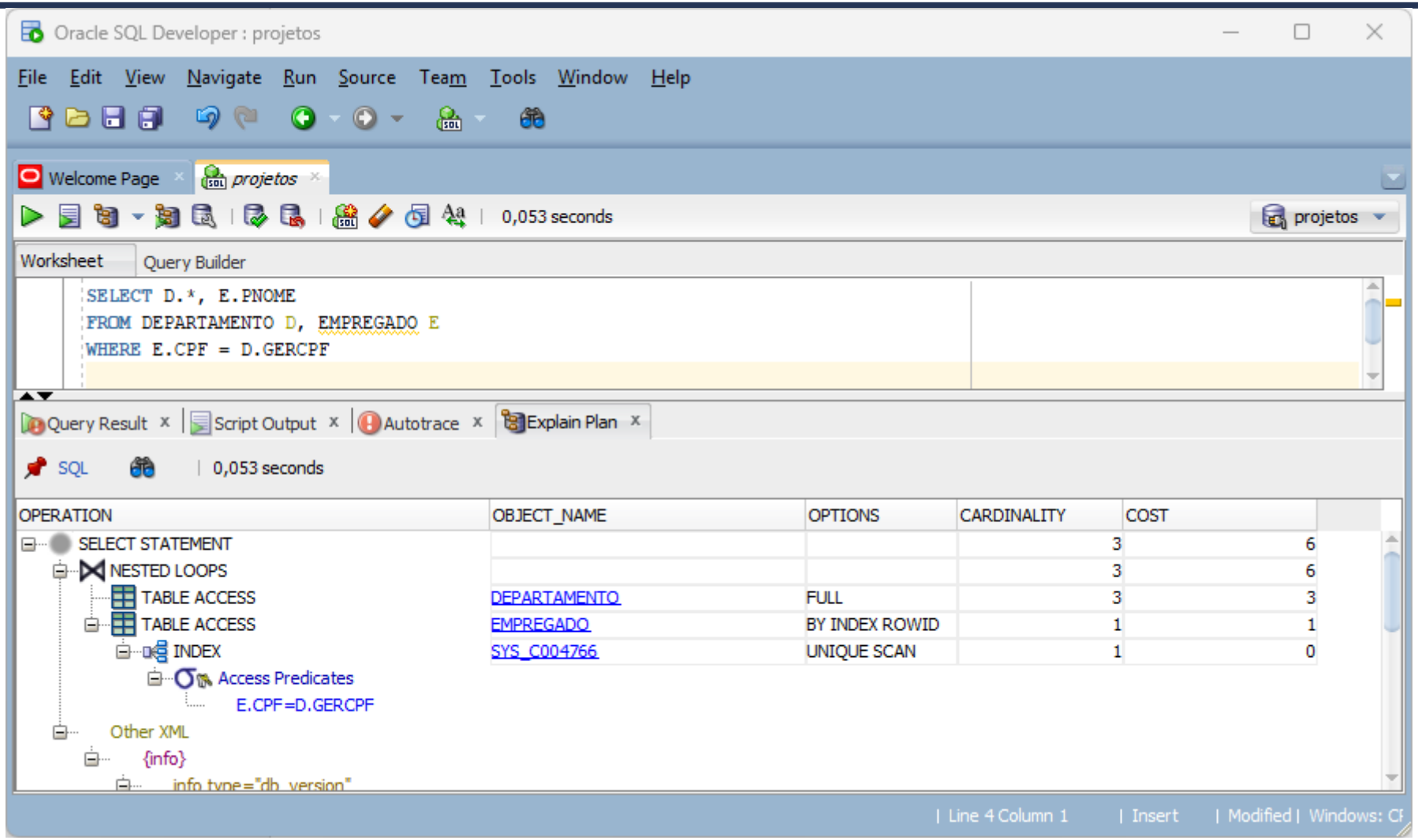
SELECT * FROM EMPREGADO E WHERE E.PNOME = 'Alicia'

Sub-consulta na projeção (lista de departamentos com seus respectivos gerentes)



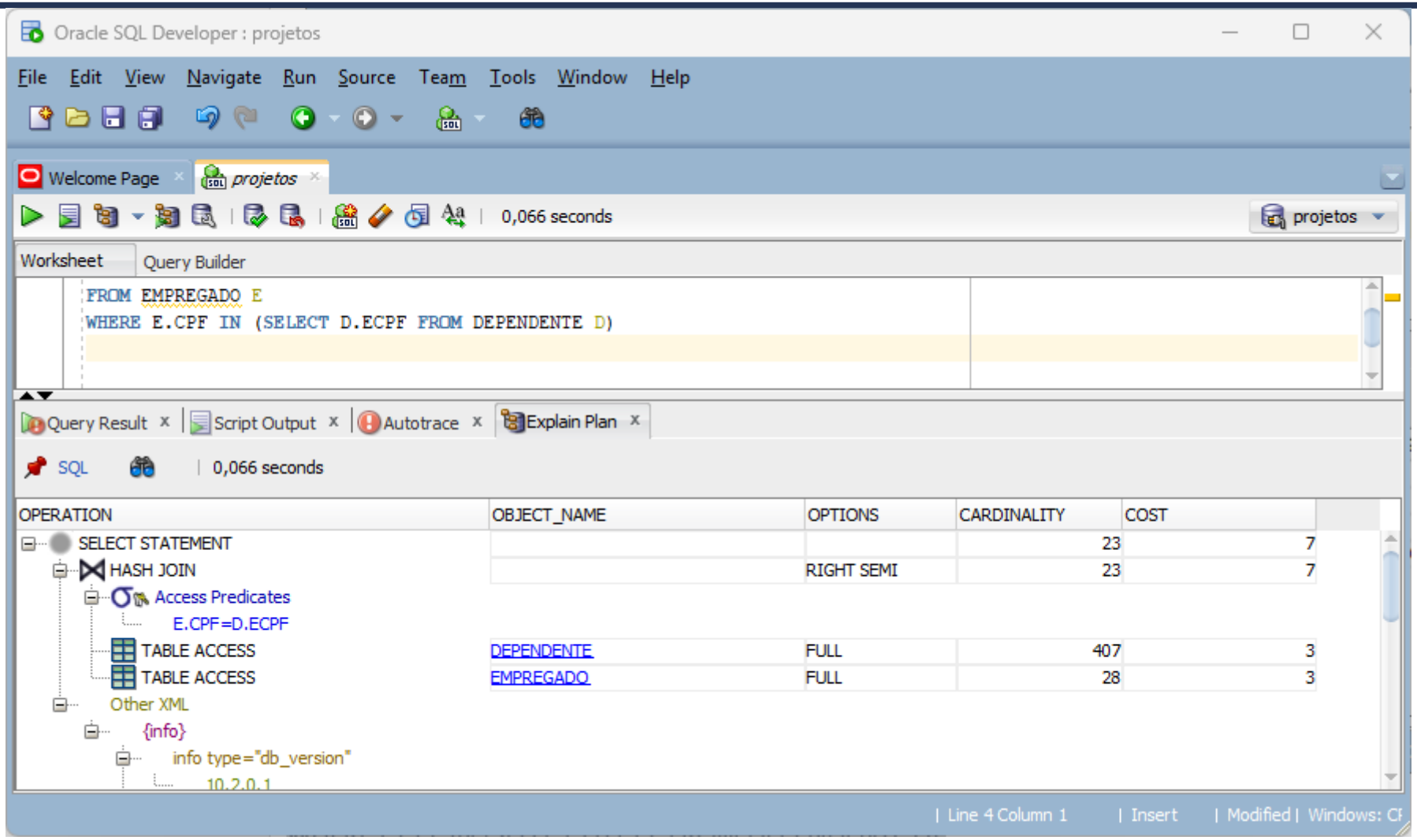

```
SELECT D.*,
   (SELECT PNOME FROM EMPREGADO E WHERE E.CPF = D.GERCPF) AS PNOME
FROM DEPARTAMENTO D
```

Consulta equivalente em junção (lista de departamentos com seus respectivos gerentes)



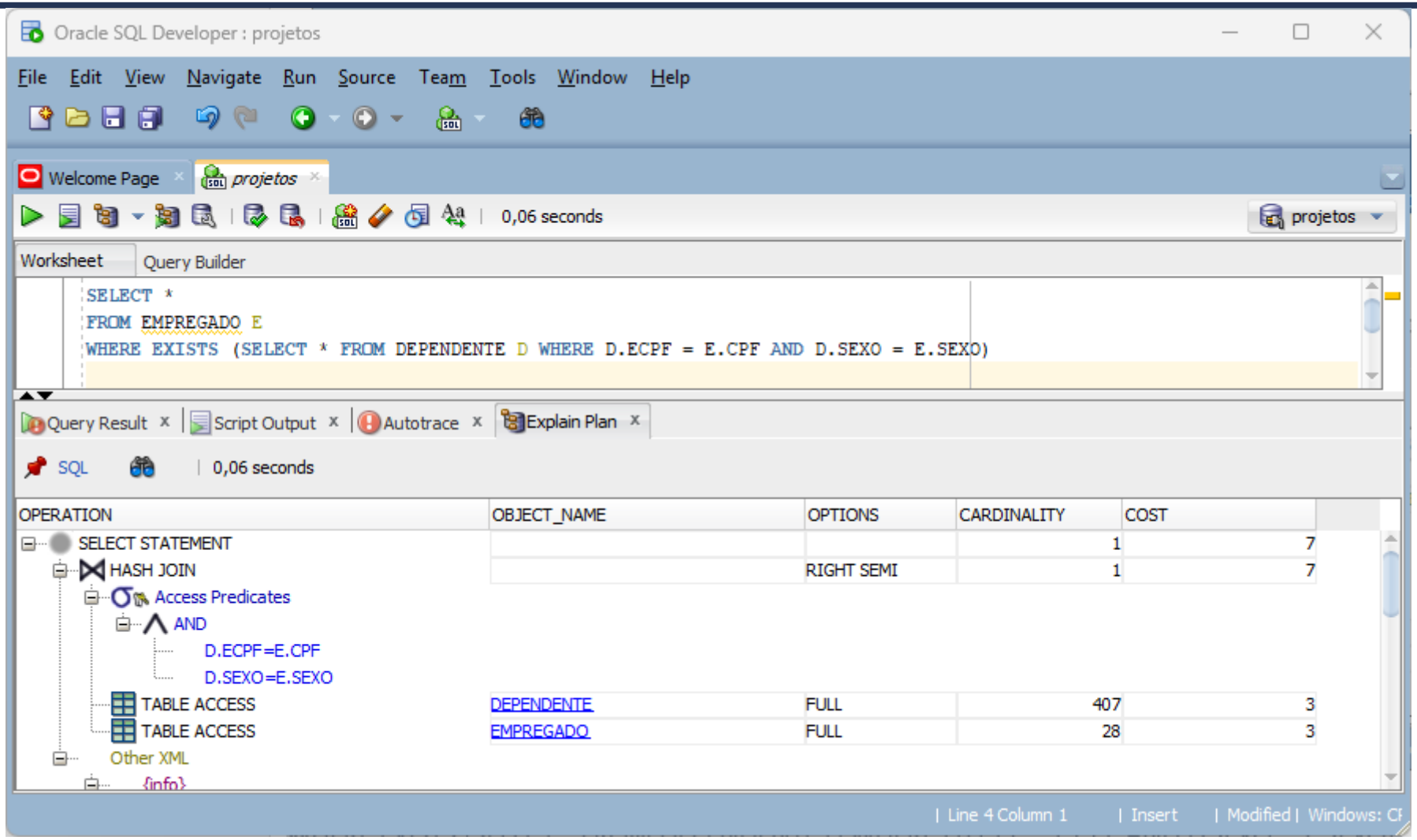

```
SELECT D.*, E.PNOME
FROM DEPARTAMENTO D, EMPREGADO E
WHERE E.CPF = D.GERCPF
```

Subconsulta não correlacionada com in (empregados com dependentes)



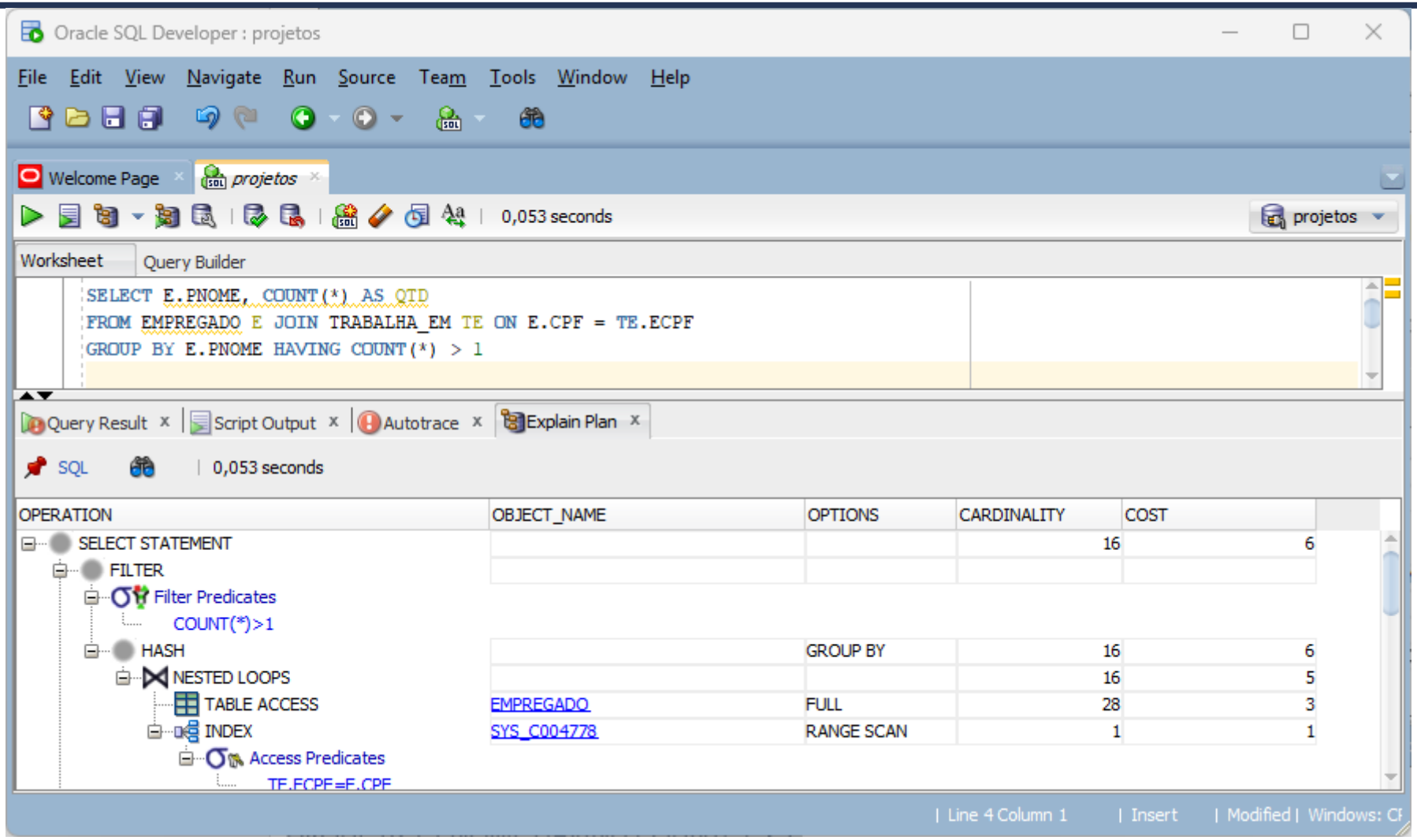

```
SELECT *
FROM EMPREGADO E
WHERE E.CPF IN (SELECT D.ECPF FROM DEPENDENTE D)
```

Subconsulta correlacionada com exists (empregados com dependentes do mesmo sexo)



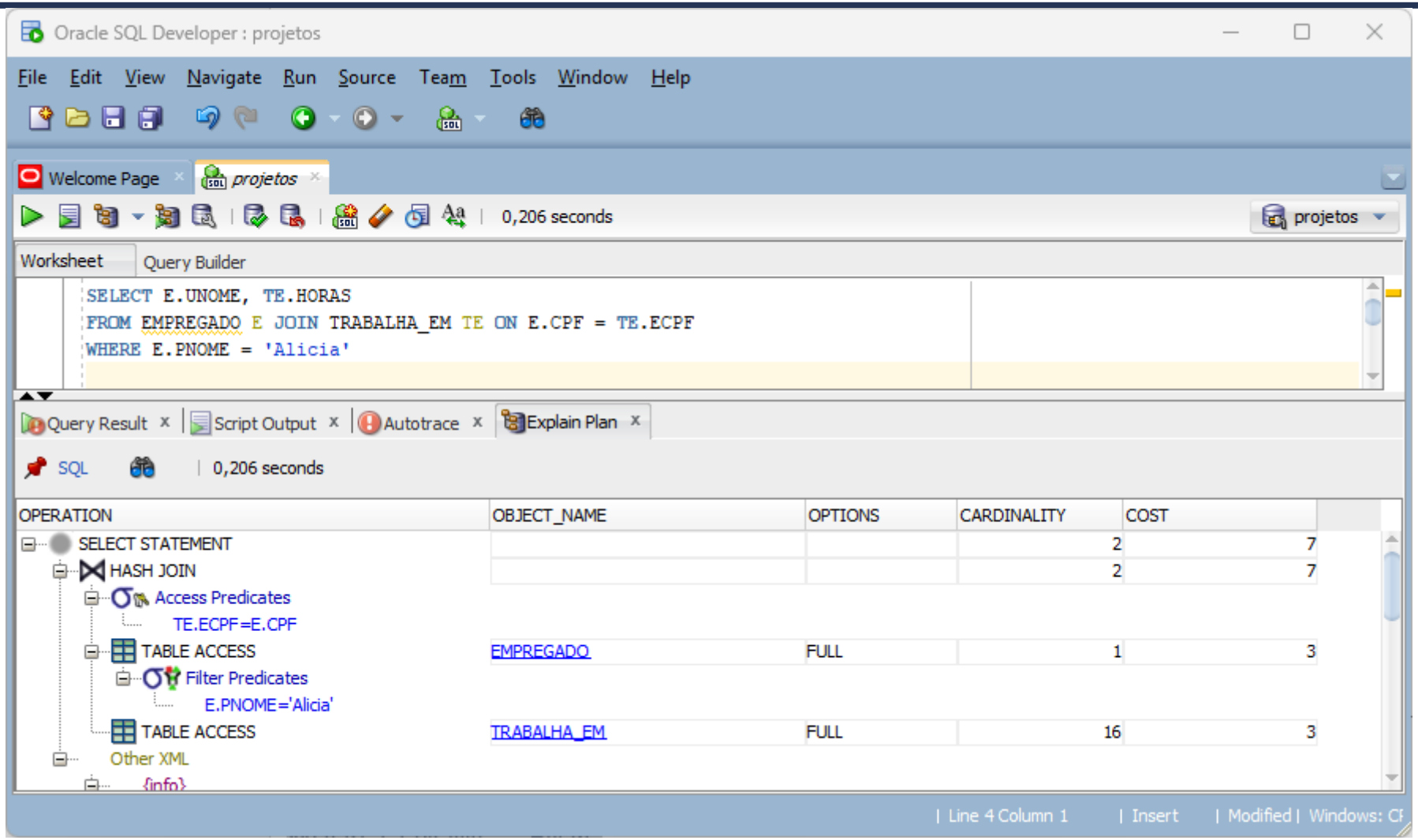

```
SELECT *
FROM EMPREGADO E
WHERE EXISTS (SELECT * FROM DEPENDENTE D WHERE D.ECPF = E.CPF AND D.SEXO = E.SEXO)
```

Consulta com agregação (quantidade de empregados que trabalham em mais de um projeto)




```
SELECT E.PNOME, COUNT(*) AS QTD
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
GROUP BY E.PNOME HAVING COUNT(*) > 1
```

Junção com busca por valor (último nome e horas trabalhadas por projeto pelo primeiro nome)

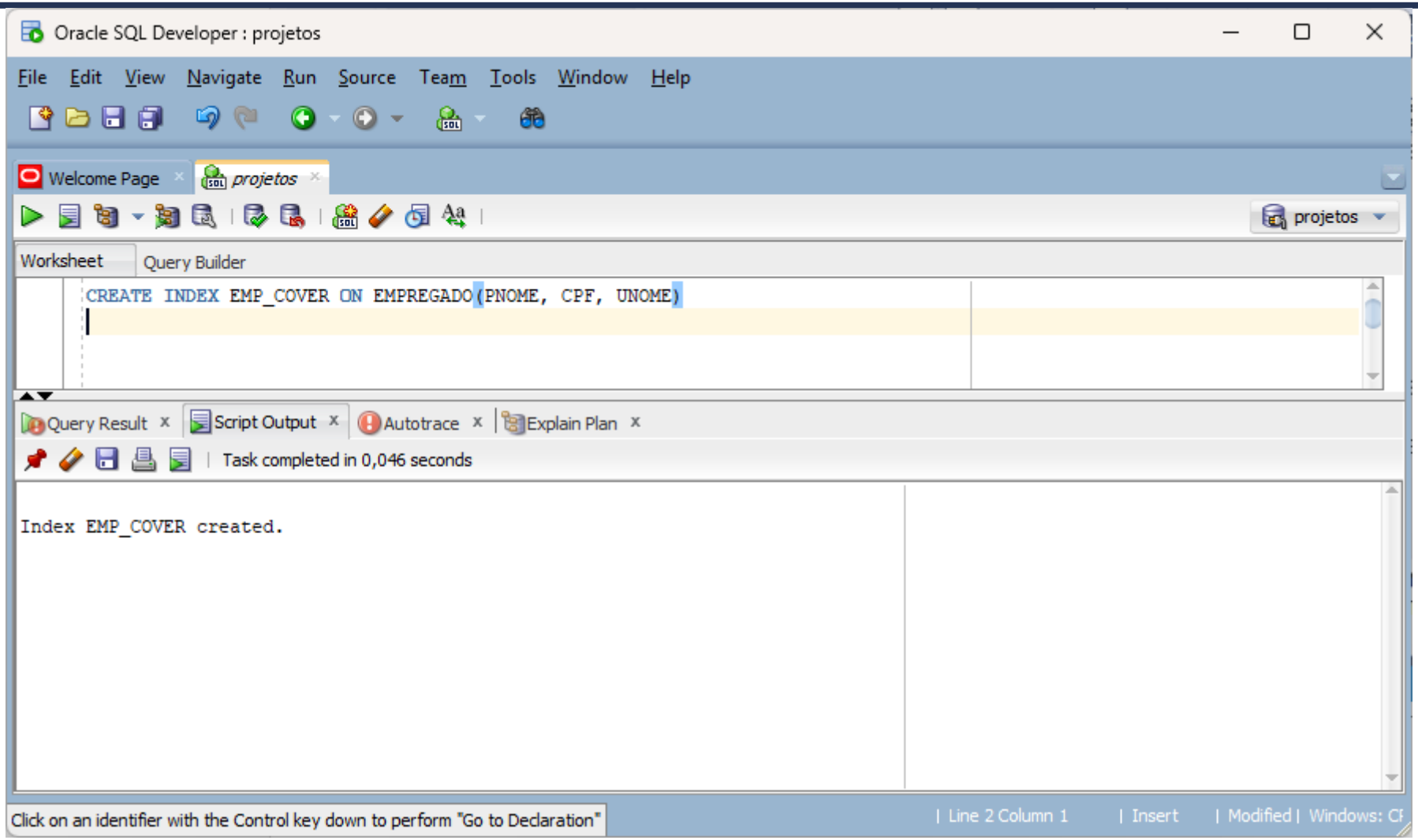



```
SELECT E.UNOME, TE.HORAS
FROM EMPREGADO E JOIN TRABALHA_EM TE ON E.CPF = TE.ECPF
WHERE E.PNOME = 'Alicia'
```

Índice de cobertura de consulta

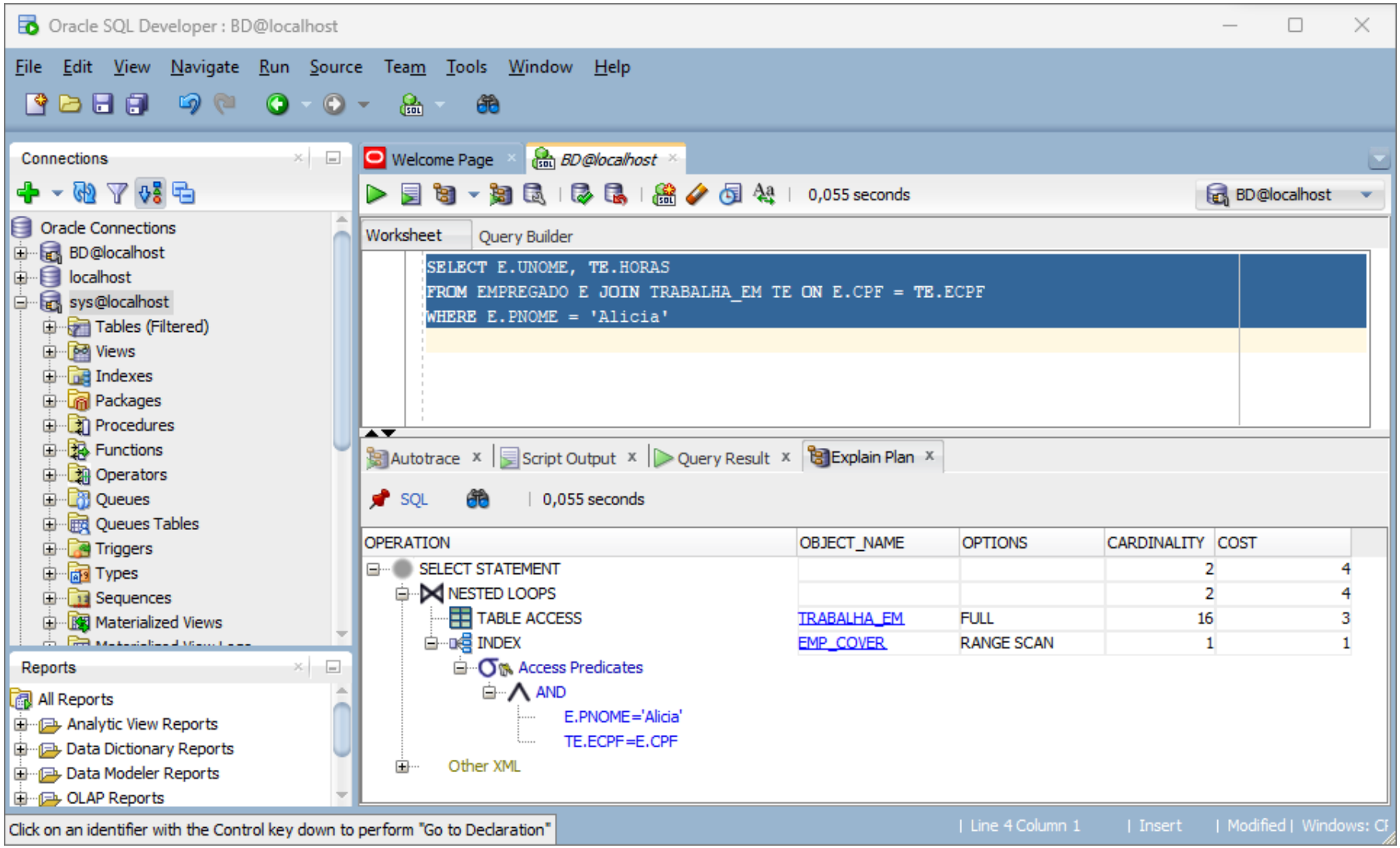
- O índice de cobertura de consulta possibilita processar a consulta sem selecionar a tabela
- Todos os atributos usados na consulta para uma Tabela T estão presentes no índice
- A ordem de criação dos campos no índice é relevante
  - Primeiro os atributos de seleção (mais restritivos primeiro)
  - Segundo os atributos usados na junção
  - Terceiro os atributos usados na projeção

Índice de cobertura de consulta



CREATE INDEX EMP_COVER ON EMPREGADO(PNOME, CPF, UNOME)

Uso do índice de cobertura de consulta (Consulta de alocação por data de nascimento com índice)



Referências