**Algebra Relacional:**

**-- Slide 10**

-- 1) πnome, data (σ Medicos.codm=Consultas.codm (Medicos ⨯ Consultas))

-- 2) πAmbulatorios.nroa,andar,nome (σAmbulatorios.nroa=Medicos.nroa ((σ andar=30 (Ambulatorios)) ⨯ (σespecialidade='ortopedia' (Medicos))))

-- 3) π Medicos.nome, Pacientes.nome, data (σ Consultas.codp=Pacientes.codp (σConsultas.codm=Medicos.codm ((σdata=date('2006-06-13') ∧ hora≥'07:00:00' ∧ hora ≤ '12:00:00' (Consultas)) ⨯ Medicos) ⨯ Pacientes))

-- 4) RF = πcpf,nome,salario (σcidade='Florianopolis' ∨ cidade='Palhoca' (Funcionarios) )

RP = πFuncionarios.nome,Funcionarios.salario, codp (σFuncionarios.cpf=Pacientes.cpf (RF ⨯ Pacientes))

RM = πcodm (σespecialidade='pediatria' (Medicos))

πnome,salario (σConsultas.codm=Medicos.codm ((σConsultas.codp=Pacientes.codp (Consultas ⨯ RP)) ⨯ RM))

----------------------------------------------------------------------------

**--OtoSlaide15**

-- 1dorho) πA1.nroa (σA1.capacidade>Ambulatorios.capacidade (ρ A1 (Ambulatorios) ⨯ (σnroa=1 (Ambulatorios))))

-- 2dorho) πEspecial.nome, Especial.cpf (σEspecial.salario<=Funcionarios.salario ∧ Especial.codf≠Funcionarios.codf (rho Especial (Funcionarios) ⨯ (σFuncionarios.cpf=1001 (Funcionarios))))

-- 3dorho) DT = πnome, data, Medicos.codm (Medicos ⨯ Consultas)

πa.nome, Medicos.nome (σa.data=Consultas.data ∧ a.codm<Medicos.codm (rho a (DT) ⨯ DT))

{

analisar 3dorho)

DT = πMedicos.nome, data, Medicos.codm (σMedicos.codm=Consultas.codm (Medicos ⨯ Consultas))

πa.nome, Medicos.nome (σa.data=Consultas.data ∧ a.codm<Medicos.codm (rho a (DT) ⨯ DT))

}

--------------------------------------

-SLIDE 21

-- 1) πnome,cpf (Pacientes) ∪ πnome,cpf (Medicos)

-- 2) πnome,cpf,idade (σ cidade = 'Florianopolis' (Pacientes)) ∪

πnome,cpf,idade (σ cidade = 'Florianopolis' (Medicos)) ∪

πnome,cpf,idade (σ cidade = 'Florianopolis' (Funcionarios))

-- 3) π nome, cpf (σ salario<500 (Funcionarios)) - πnome,cpf (Pacientes)

-- 4) πnroa (Ambulatorios) - πnroa (Medicos)

-- 5) π nome, cpf (σ cidade='Florianopolis' (Funcionarios)) ∩ πnome,cpf (Pacientes)

-- (outra forma) 5) πFuncionarios.nome, Funcionarios.cpf (σFuncionarios.cpf=Pacientes.cpf (σcidade='Florianopolis' (Funcionarios) ⨯ Pacientes))

-- 6) πnome,cpf (σespecialidade='pediatria' ∧ nroa≠101 ∧ nroa≠102 (Medicos)) ∩

πnome,cpf (σdoenca='gastrite' (Pacientes))

-- (outra forma) 6) πMedicos.nome, Medicos.cpf (σMedicos.cpf=Pacientes.cpf (πnome,cpf (σespecialidade='pediatria' ∧ nroa≠101 ∧ nroa ≠ 102 (Medicos)) ⨝ Medicos.cpf = Pacientes.cpf (πcpf (σdoenca='gastrite' (Pacientes)))))

-------------------------------------------

Slide 26

-- 1) πAmbulatorios.nroa,capacidade,nome (σandar=50 (Ambulatorios) ⨝ Medicos)

-- 2) πnome, salario (σcidade = 'Florianopolis' ∨ cidade = 'Palhoca' (Funcionarios) ⨝ Pacientes ⨝ (σ data = date('2006-10-20') (Consultas)))

-- outra forma 2) πnome, salario ((πnome, salario, cpf (σcidade = 'Florianopolis' ∨ cidade = 'Palhoca' (Funcionarios) ⨝ (πcodp, cpf (Pacientes)))) ⨝ (σ data = date('2006-10-20') (Consultas)))

-- 3) πnroa,andar (Ambulatorios) - (πAmbulatorios.nroa,andar (Ambulatorios ⨝ Medicos))

-- outra forma (PETER) 3) πAmbulatorios.nroa, Ambulatorios.andar (σMedicos.codm = null (πnroa, andar Ambulatorios ⟕ πcodm, nroa Medicos))

-- outra forma 3) πAmbulatorios.nroa, andar (πAmbulatorios.nroa, andar (Ambulatorios ⟕ Medicos) - πAmbulatorios.nroa, andar (Ambulatorios ⨝ Medicos))

-- 4) πA2.nroa (ρA1 (σnroa=101 (Ambulatorios)) ⨝A1.andar=A2.andar ∧ A2.capacidade>A1.capacidade ρA2 (Ambulatorios))

-------------------------------------------

Slide 29

-- 1) (Medicos) ⟕ (Consultas)

-- 2) πAmbulatorios.nroa,codm, nome (Ambulatorios ⟕ Medicos)

-- 3) pi Medicos.cpf, Medicos.nome, Pacientes.cpf, Pacientes.nome ((Medicos) ⟗ Medicos.codm = Consultas.codm ((Pacientes) ⟕ (Consultas)))

-- outra forma 3) π Medicos.cpf, Medicos.nome, Pacientes.cpf, Pacientes.nome (π cpf, nome, codp (Medicos⟕Consultas) ⟗Consultas.codp=Pacientes.codp Pacientes)

(Questão 3 Slide 26 refeito) - Submissao: 03:19 da manhã de segunda-feira :’(

-- 3) πAmbulatorios.nroa, Ambulatorios.andar (σMedicos.codm = null (πnroa, andar Ambulatorios ⟕ πcodm, nroa Medicos))

Slide 32

-- 1) (π codp, codm (Consultas)) ÷ (πcodm (Medicos))

--2) RM = πcodm, codp (Consultas) ÷ πcodp (Pacientes)

πnome, cpf (RM ⨝ Medicos)

-- outra forma 2) πcpf, nome (πcodm, cpf, nome, codp (Consultas ⨝ Medicos) ÷ πPacientes.codp (Pacientes))

-- 3) NAO EH ASSIM

ORT = (pi codm (Medicos ⨝ Ambulatorios.andar=1∧Medicos.especialidade='ortopedia' Ambulatorios))

pi nome, cpf (Pacientes ⨝ ORT)

-- 3) πnome, cpf, codm (Pacientes⨝ Consultas) ÷ πcodm (σespecialidade='ortopedia' (Medicos) ⨝ σandar=1 (Ambulatorios))

-- 4) πnroa, andar, codm (Ambulatorios⨝Medicos) ÷ (πcodm (σespecialidade = 'ortopedia' (Medicos)))

SLIDE 35

-- 2) Pacientes ∪ πcodf, nome, idade, cidade, cpf (σcpf=1000 (Funcionarios)) ⨯ {('hepatite')}

-- 3)σdata = date('2006-08-10') (σcodm=37 ∧ data=date('2006-09-29') (Consultas))

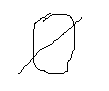
-- 4) πAmbulatorios.nroa, capacidade, andar (σnome ≠ null (Ambulatorios ⟕ Medicos))

Lista para entregar

-- 1) πid\_cd, titulo (σtipo='coletanea' ∨ tipo='trilha sonora' ∧ duracao>90 ∧ duracao<120 (CDs))

-- 2) πInterpretes.nome (σCDs.ano=2013 ∧ Interpretes.nacionalidade='Brasil' ∧ Interpretes.tipo\_inter='grupo' (σCDs.id\_inter=Interpretes.id\_inter (CDs⨯Interpretes)))

-- 2) outraforma (melhor): πnome (σCDs.ano=2013 ∧ Interpretes.nacionalidade='Brasil' ∧ Interpretes.tipo\_inter='grupo' (CDs⨝Interpretes))

-- 3) π CDs.id\_cd, titulo, Interpretes.nome, Interpretes.tipo\_inter (σpreco>30 (σCDs.id\_inter=Interpretes.id\_inter (CDs⨯Interpretes)))

-- 4) πExemplares.codBarras, Funcionarios.funNome,funID (σdata=date('2011-04-30')∨data=date('2014-05-01')∧formaPGTO≠'cartao' (Vendas⨝Funcionarios⨝Exemplares))

-- 5) σstatus='ok' (Exemplares)⨝CDs

-- 6) πa.funID,a.funNome, Funcionarios.funID, Funcionarios.funNome (σa.turno=Funcionarios.turno ∧ a.funID<Funcionarios.funID (ρ a (Funcionarios) ⨯ Funcionarios))

-- 7) πnome, id\_inter ((σnacionalidade≠'Brasil' (Interpretes)) ⨝ (σtipo='coletanea' ∨ tipo='interprete' (CDs)))

-- 8) πfunID, funNome (πfunID, funNome, id\_cd ((σturno='noturno' (Funcionarios) ⨝ Vendas)) ÷ (πid\_cd (σInterpretes.nome='Ivete Sangalo' (Interpretes⨝CDs))))

-- 9) pao = πid\_cd, titulo (CDs)

margarina = πc1.id\_cd, c1.titulo (ρc1 (CDs) ⨝c1.preco<c2.preco ρc2 (CDs))

sanduba = pi id\_cd (pao - margarina)

refri = pi id\_cd, funID, salario (Funcionarios ⨝ Vendas)

del salario ← salario\*1.1 (refri ⨝ sanduba)

-- 10) pocahontas = pi salario, turno (σcpf=22334455667 (Funcionarios))

Funcionarios ← Funcionarios ∪ ( { (25, 'Pedro Souza', 11223344556, 0, 9999999999, 'rua X, 333, Joinville') } ⨯ pocahontas)

-- 7) “certa”

R1=πid\_inter (σtipo='coletanea' (CDs))

R2=πid\_inter (σtipo='interprete' (CDs))

R3= πid\_inter, nome (σnacionalidade≠'Brasil' (Interpretes))

πa.id\_inter, a.nome ((ρ a (R3) ⨝ a.id\_inter=b.id\_inter rho b (R1)) ⨝ a.id\_inter=Interpretes.id\_inter (Interpretes))

-- 8) “certa”

r1= πid\_cd (CDs ⨝ CDs.id\_inter = Interpretes.id\_inter (πid\_inter (σnome='Ivete Sangalo' (Interpretes))))

r2= πfunID, funNome (σturno='noturno' (Funcionarios))

r3 = πa.funID, funNome, id\_cd (ρ a (r2) ⨝ a.funID=Vendas.funID Vendas)

r3÷r1

SQL-TOP

SLIDE 34

1. SELECT nome, cpf from medicos where especialidade <> 'traumatologia' or idade < 40
2. select \* from consultas where data > date('2006-06-19') and hora between '12:00:00' and '18:00:00'







11. select round(avg(idade),2) as MediaIdade, count(distinct(nroa)) as AmbAtPorEles from medicos
12. select codf, nome, salario\*0.8 as salarioLiquido from funcionarios
14. select nome, cpf from funcionarios where cpf::text not like '%00000%';
15. select nome, especialidade from medicos where nome like '\_o%o'
16. select codp, nome from pacientes where idade>25 and doenca in ('tendinite', 'fratura', 'gripe', 'sarampo')

SLIDE 43

1. select m.nome, m.cpf from medicos m natural join pacientes p
2. select codf, f.nome, codm, m.nome from funcionarios f join medicos m on f.cidade=m.cidade
3. select p.codp, nome from pacientes p natural join consultas where hora>('14:00:00')
4. select a.nroa, andar from ambulatorios a natural join medicos where especialidade='ortopedia'
5. select nome, cpf from pacientes p join (select codp from consultas where data between '2006-06-14' and '2006-06-16')c on p.codp=c.codp;
6. select m.nome, m.idade from medicos m natural join consultas c join pacientes p on c.codp=p.codp where p.nome='Ana'
7. (deve estar errada) select m2.codm, m2.nome from medicos m1 join medicos m2 on m1.nome='Pedro' and m1.nroa=m2.nroa join consultas c on c.codm=m2.codm where data='2016-06-14' and m1.codm!=m2.codm;
8. select p.nome, p.cpf, p.idade from pacientes p natural join consultas c join medicos m on m.codm=c.codm where especialidade='ortopedia' and date\_part('day', data)<16;
9. prof vai colocar o gabarito
10. select a.\*, m.codm, m.nome from ambulatorios a left join medicos m on a.nroa=m.nroa;

EXERCÍCIOS BÔNUS - FOTOS NA PASTA

BONUS: select nome from pacientes p natural join consultas c where p.codp not in

(select c.codp from medicos m natural join consultas c

where especialidade not in ('ortopedia','traumatologia'));

SLIDE 63

**IN:**

1. select nome, cpf from medicos where cpf in (select cpf from pacientes)
2. select codp, cpf from pacientes where codp in (select codp from consultas where hora > '14:00:00')
3. **nao utilizando in:** SELECT m.nome, m.idade from medicos m natural join consultas c join pacientes p on c.codp=p.codp where p.nome='Ana';   
   **utilizando in:** select nome, idade from medicos where codm in (select codm from consultas where codp in (select codp from pacientes where nome='Ana'))
4. select nroa, andar from ambulatorios where nroa not in (select nroa from medicos)
5. **select distinct nome, cpf from pacientes natural join consultas where codp not in (select codp from consultas where date\_part('day',data) >= 16)** (pacientes que não tem nenhum horário marcado de consulta não aparecerão tbm)

**ANY/ALL:**

1. select nroa, andar from ambulatorios where capacidade > ANY (select capacidade from ambulatorios)
2. select nome, idade from medicos where codm=any (select codm from consultas where codp = any (select codp from pacientes where nome='ana'))
3. select nome, idade from medicos where idade <= ALL (select idade from medicos)
4. select nome, cpf from pacientes where codp < ALL (select codp from consultas where hora < ANY (select hora from consultas where data = '12-11-2006'))

resposta da prof: select distinct nome, cpf from pacientes natural join consultas where hora < ALL (select hora from consultas where data = '12-11-2006')

1. select nome, cpf from medicos where nroa <> all (select nroa from ambulatorios where capacidade > all(select capacidade from ambulatorios where andar = 2))

**EXISTS:**

1. select nome, cpf from medicos m where EXISTS (select \* from pacientes where cpf=m.cpf)
2. solução chuncho: select nome, idade from medicos m where exists (select \* from (consultas natural join pacientes) where nome='Ana' and m.codm=codm)  
     
   solução não chuncho: select nome, idade from medicos m where exists (select \* from pacientes p where nome = 'Ana' and exists (select \* from consultas c where c.codp=p.codp and m.codm=c.codm))
3. select nroa from ambulatorios a where not exists (select \* from ambulatorios aa where aa.capacidade>a.capacidade)
4. select nome, cpf from medicos m where not exists (select \* from pacientes p where not exists (select from consultas c where c.codp=p.codp and m.codm=c.codm))
5. select nome, cpf from medicos m where not exists (select \* from pacientes p where cidade='Florianopolis' and not exists (select from consultas c where c.codp=p.codp and m.codm=c.codm))

SLIDE 74

**FROM:**

1. select c.\* from consultas c natural join (select codm from medicos where nome = 'Maria') as M
2. select p.nome, p.codp from pacientes p natural join (select codp from consultas where hora>'14:00:00') as bumbum
3. select distinct p.nome, p.cidade from pacientes p natural join (select codp from consultas natural join (select codm from medicos where especialidade='ortopedia') as tschicaplau) as bumbum
4. select p.nome, p.cpf from (select nome, cpf, codp from pacientes where cidade='Florianopolis') as p except select p.nome, p.cpf from pacientes p natural join consultas c natural join (select codm from medicos where nome='Joao') as truco

**ORDER BY e GROUP BY:**

1. select \* from funcionarios order by salario desc, idade asc limit 3
2. select nome, nroa, andar from medicos natural join ambulatorios order by andar asc
3. select x.nome, p.nome from (medicos natural join consultas) as x inner join pacientes p on p.codp=x.codp order by data asc, hora asc limit 3 offset 2
4. select idade, count(\*) from medicos group by idade
5. select data, count(\*) from (select data from consultas where hora > '12:00:00') as cu group by data
6. select andar, round(avg(capacidade), 2) from ambulatorios group by andar
7. select andar from ambulatorios group by andar having avg(capacidade) >= 40
8. select nome from medicos natural join consultas

group by codm, nome

having count(\*)>1

**ATUALIZAÇÕES:**

1. update consultas set hora = '19:00:00' where codp in = (select codp from pacientes where nome = ‘Ana’)
2. delete from pacientes where codp not in (select codp from consultas)
3. update consultas set data = '2006-11-21' where hora < '12:00:00' and codm in (select codm from medicos where nome = 'Pedro')
4. update ambulatorios set andar = (select andar from ambulatorios where nroa=1), capacidade = 2\* (select max(capacidade) from ambulatorios) where nroa = 4
5. insert into medicos (codm, nome, idade, cpf, cidade, especialidade, nroa)

(select \* from (select max(codm) + 1 from medicos) as M,

(select nome, idade, cpf, cidade from funcionarios where codf=3) as F,

(select especialidade, nroa from medicos where codm=2) as M2)

LISTA PARA ENTREGAR

1. select x.id, x.nome from (select \* from funcionarios f join clientes c on f.id=c.id where turno='noturno') as x join reservas r on x.id=r.cliente where r.dataR='01-10-2006'
2. select pai.nome , pai.endereco, filho.nome from Clientes as pai join Clientes as filho on pai.ID = filho.responsavel where pai.cidade = 'Florianópolis'
3. select c.nome, c.endereco from clientes c join locacoes l on c.id=l.cliente where dataPD>dataD
4. (select nome, cidade, endereco from Funcionarios where turno = 'M' or turno = 'T') union (select nome, cidade, endereco from Clientes join Reservas r ON ID = r.cliente)
5. select x.cliente, x.nome, x.fone (select \* from clientes c1 join locacoes l1 on c1.id=l1.cliente join copias e1 on l1.filme=e1.filme where midia='vhs') as x join (select \* from clientes c2 join locacoes l2 on c2.id=l2.cliente join copias e2 on l2.filme=e2.filme where midia='dvd') as y on x.cliente=y.cliente
6. select a.ID , a.filme, ID, filme from Copias as a join Copias on a.filme = filme where a.ID > ID
7. select client.ID, nome, fone from Clientes cliient where client.ID not in (select cliente from Locacoes join (select ID from Filmes join (select ID from Estilos where nome != 'acao' and nome != 'suspense') as tipos on Filmes.estilo = tipos.ID) as \_filmes) on filme = \_filmes.ID

----  
Questao 1: select x.id, x.nome from (select \* from funcionarios f join clientes c on f.id=c.id where turno='noturno') as x join reservas r on x.id=r.cliente where r.dataR='01-10-2006'  
  
----  
Questao 2: select pai.nome , pai.endereco, filho.nome from Clientes as pai join Clientes as filho on pai.ID = filho.responsavel where pai.cidade = 'Florianópolis'  
  
----  
Questao 3: select c.nome, c.endereco from clientes c join locacoes l on c.id=l.cliente where dataPD>dataD  
  
----  
Questao 4: (select nome, cidade, endereco from Funcionarios where turno = 'M' or turno = 'T') union (select nome, cidade, endereco from Clientes join Reservas r ON ID = r.cliente)  
  
----  
Questao 5: select x.cliente, x.nome, x.fone (select \* from clientes c1 join locacoes l1 on c1.id=l1.cliente join copias e1 on l1.filme=e1.filme where midia='vhs') as x join (select \* from clientes c2 join locacoes l2 on c2.id=l2.cliente join copias e2 on l2.filme=e2.filme where midia='dvd') as y on x.cliente=y.cliente  
  
----  
Questao 6: select a.ID , a.filme, ID, filme from Copias as a join Copias on a.filme = filme where a.ID > ID  
  
----  
Questao 7: select client.ID, nome, fone from Clientes cliient where client.ID not in (select cliente from Locacoes join (select ID from Filmes join (select ID from Estilos where nome != 'acao' and nome != 'suspense') as tipos on Filmes.estilo = tipos.ID) as \_filmes) on filme = \_filmes.ID

----

CORREÇÃO:

1. -- select f.id, f.nome from funcionarios f join clientes c on f.cpf=c.cpf join reservas r on r.cliente=c.id and dataRes='2006-10-01' where f.turno='N'  
     
   -- select f.id, f.nome from funcionarios f, clientes c, reservas r where f.turno='N' and f.cpf=c.cpf and r.cliente=c.id and dataRes='2006-10-01'  
     
   -- select f.id, f.nome from funcionarios f where turno='N' and cpf in (select cpf from clientes where id in (select cliente from reservas where dataRes='2006-10-01'))  
     
   -- test: select f.id, f.nome from funcionarios f where turno='N' and cpf=any (select cpf from clientes where id=any (select cliente from reservas where dataRes='2006-10-01'))  
     
   -- select f.id, f.nome from funcionarios f where turno='N' and exists (select \* from clientes c where exists (select \* from reservas r where dataRes='2006-10-01' and f.cpf=c.cpf and c.id=r.cliente))  
     
   -- select f.id, f.nome from (select id, nome, cpf from funcionarios where turno='N') as f join clientes c on f.cpf=c.cpf join (select cliente from reservas where dataRes='2006-10-01') as r on c.id=r.cliente
2. select c1.nome, c1.endereco, c2.nome from clientes c1 left join clientes c2 on c1.id=c2.responsavel where c1.cidade='Florianopolis'
3. trocar o natural join: select c.nome,l.endereco from clientes c join locacoes l on c.id=l.cliente and dataD>dataPD natural join copias cp where cp.midia='DVD'
4. select nome, cidade, endereco from funcionarios where turno!='N' UNION select nome, cidade, endereco from clientes c join reservas r on c.id=r.cliente
5. select c.id, nome, fone from clientes c join locacoes l on c.id=l.cliente join copias cp on cp.id=l.id and cp.filme=l.filme where midia='VHS' intersect select c.id, nome, fone from clientes c join locacoes l on c.id=l.cliente join copias cp on cp.id=l.id and cp.filme=l.filme where midia='DVD'
6. select cp1.id, cp1.filme, cp2.id, cp2.filme from copias cp1 join copias cp2 on cp1.filme=cp2.filme and cp1.id<cp2.id
7. select c.id, c.nome, c.fone from clientes c join locacoes l on c.id=l.id join filmes f on l.filme=f.id join estilos e on e.id=f.estilo where e.nome in ('acao','suspense')and not exists (select \* from locacoes l2 join filmes f2 on l2.filme=f2.id join estilos e2 on e2.id=f2.estilo where e2.nome not in ('acao','suspense') and l2.cliente=c.id)  
     
   outra forma: select c.id, c.nome, c.fone from clientes c join locacoes l on c.id=l.cliente except select c.id, c.nome, c.fone from clientes c join locacoes l on c.id=l.cliente join filmes f on f.id=l.filme join estilos e on e.id=f.estilo where e.nome not in ('acao','suspense')

CORREÇÃO DA PROVA

1.

1. select r1.cidadeOrigem, r1.cidadeDestino, r2.cidadeOrigem, r2.cidadeDestino from rotas r1 join rotas r2 on r1.codEmpresa=r2.codEmpresa and r1.codRota < r2.codRota
2. -- select nome, cpf from destinatarios natural join pacotes **except** select d.nome, d.cpf from destinatarios d natural join pacotes p natural join rotas r join transportadoras t on t.codEmpres=r.codEmpresa where t.nome!='Catarinense'  
     
   -- select nome, cpf from destinatarios d natural join pacotes where not exists (select \* from pacotes p natural join rotas r join transportadoras t on t.coEmpresa=r.codEmpresa where t.nome!='Catarinense' and p.codDest=d.codDest)  
     
   -- select nome, cpf from destinatarios d natural join pacotes where codDest not in (select p.codDest from pacotes p natural join rotas r join transportadoras t on t.coEmpresa=r.codEmpresa where t.nome!='Catarinense')
3. select t1.nome, t1.autorizacao, t2.nome from transportadoras t1 left join transpotadoras t2 on t1.codEmpresaResponsavel = t2.codEmpresa and t1.codEmpresa!=t2.codEmpresa
4. -- select d.codDest, nome from destinatarios d natural join pacotes where peso = (select max(peso) from pacotes)  
     
   -- select d.codDest, nome from destinatarios d natural join pacotes order by peso desc limit1  
     
   -- select d.codDest, nome from destinatarios d natural join pacotes **except** (select d.codDest, nom from destinatiors d natural join pacotes p1 join pacotes p2 on p1.peso<p2.peso)