Lab05

Grupo - **P2G2**

Membros	Número Mecanográfico
David Araújo	93444
Miguel Nogueira	93082

Exercício 1

a)

> π project.Pname, project.Pnumber, project.Plocation, project.Dnum, employee.Fname, employee.Lname, employee.Ssn works_on ⋈ Pno = Pnumber project ⋈ Ssn = Essn employee

b)

> π a.Fname, a.Minit, a.Lname, a.Ssn, a.Bdate, a.Address, a.Sex, a.Salary, a.Super_ssn, a.Dno ρ a employee ⋈ b.Ssn = a.Super_ssn ρ b π b.Ssn σ b.Fname = 'Carlos' and b.Lname = 'Gomes' ρ b employee

c)

> π project.Pname, a.Total project ⋈ a.Pno = Pnumber ρ a π Pno, Total γ Pno; SUM(Hours)→Total works_on

d)

> π Fname, Minit, Lname employee ⋈ employee.Ssn = a.Essn ρ a π Essn works_on ⋈ works_on.Hours > 20 ρ b π Pnumber σ Pname = 'Aveiro Digital' project

e)

> π employee.Fname, employee.Minit, employee.Lname employee ⋉ Ssn = Essn works_on

f)

> π department.Dname, AVGSalary, employee.Sex γ department.Dname, employee.Sex; AVG(employee.Salary)→AVGSalary department ⋈ employee.Dno = department.Dnumber employee

g)

> π employee.Fname, employee.Minit, employee.Lname, e.Ndependents σ e.Ndependents > 2 employee ⋈ employee.Ssn = e.Essn ρ e π dependent.Essn, Ndependents γ Essn; COUNT(Essn)→Ndependents dependent

h)

> π emp.Fname, emp.Minit, emp.Lname, emp.Ssn, department.Dname department ⋈ department.Mgr_ssn = emp.Ssn ρ emp π Fname, Minit, Lname, Ssn σ Essn = null employee ⟕ Essn = Ssn dependent

i)

> π wpre.Fname, wpre.Minit, wpre.Lname, wpre.Pname, wpre.Plocation, dept_location.Dlocation σ dept_location.Dlocation ≠ 'Aveiro' dept_location ⋈ wpre.Dno = dept_location.Dnumber ρ wpre π employee.Fname, employee.Minit, employee.Lname, employee.Dno, wpr.Pname, wpr.Plocation employee ⋈ wpr.Essn = employee.Ssn ρ wpr π Essn, proj.Plocation, proj.Pname works_on ⋈ works_on.Pno = proj.Pnumber ρ proj π Pnumber, Plocation, Pname σ Plocation = 'Aveiro' project

Exercício 5.2

a)

> π nif (fornecedor) - π encomenda.fornecedor (encomenda ⋈ (nif = fornecedor) fornecedor)

b)

> γ produto.nome; avg(item.unidades) -> unidades π item.numEnc, produto.codigo, produto.nome, item.unidades (produto ⋈ codProd = codigo item)

c)

> γ avg(num_produtos) -> media_num_prod_por_encomenda γ item.numEnc; count(produto.codigo) -> num_produtos (item ⋈ codProd = codigo produto)

d)

> π fornecedor.nif, fornecedor.nome, produto.codigo, produto.nome, item.unidades (produto ⋈ item.codProd = codigo (fornecedor ⋈ encomenda.fornecedor = nif (item ⋈ numero = numEnc encomenda)))

Exercício 5.3

a)

> π paciente.numUtente, paciente.nome, paciente.dataNasc, paciente.endereco σ prescricao.numUtente = null paciente ⟕ paciente.numUtente = prescricao.numUtente prescricao

b)

> π medico.especialidade, Num γ especialidade; COUNT(numSNS)→Num medico ⋈ numMedico = numSNS prescricao

c)

> π prescricao.farmacia, Num γ farmacia; COUNT(farmacia)→Num σ farmacia ≠ null prescricao

d)

> farmaceutica ⋉ far.numRegFarm = farmaceutica.numReg ρ far π farmaco.numRegFarm, farmaco.nome, farmaco.formula σ farmaco.numRegFarm = 906 and presc_farmaco.numRegFarm = null farmaco ⋊ farmaco.numRegFarm = presc_farmaco.numRegFarm and farmaco.nome = presc_farmaco.nomeFarmaco presc_farmaco

e)

> π prescricao.farmacia, presc_farmaco.numRegFarm, Num γ farmacia, numRegFarm; COUNT(numRegFarm)→Num σ farmacia ≠ null presc_farmaco ⋈ prescricao.numPresc = presc_farmaco.numPresc prescricao

d)

> γ farmacia.nome, farmaceutica.nome; count(presc_farmaco.nomeFarmaco) -> num_farmacos π farmacia.nome, farmaceutica.nome, presc_farmaco.nomeFarmaco (farmaceutica ⋈ presc_farmaco.numRegFarm = numReg (farmacia ⋈ prescricao.farmacia = nome π prescricao.farmacia, presc_farmaco.numRegFarm, presc_farmaco.nomeFarmaco (prescricao ⋈ numPrescX = numPresc (ρ numPrescX←numPresc (presc_farmaco)))))