5.1

a)

π Pname, Fname, Minit, Lname, Ssn (project ⨝ Pnumber=Pno works\_on ⨝ Essn=Ssn employee)

b)

π Fname, Minit, Lname (π Fname, Minit, Lname, Super\_ssn (employee) ⨝ Super\_ssn=Ssn (πSsn (σ Fname='Carlos' and Minit='D' and Lname='Gomes' (employee))))

c)

π Pname, totalhours (γ Pname; suma(Hours) -> totalhours (project ⨝ Pnumber=Pno works\_on))

???

d)

π Fname, Minit, Lname, Ssn (π Fname, Minit, Lname, Ssn (σ Dno=3 (employee)) ⨝ Ssn=works\_on.Essn (πPno, Essn (σ Hours>20 (works\_on)) ⨝ Pno=Pnumber π Pnumber (σ Pname='Aveiro Digital' (project))))

e)

π Fname, Minit, Lname (σ Essn=null (π Fname, Minit, Lname, Ssn (employee) ⟕ Ssn=Essn works\_on))

f)

π Dname, avg\_salary (γ Dname; avg(Salary) -> avg\_salary (π Dnumber, Dname (department) ⨝ Dnumber=Dno π Salary, Dno (σ Sex='F' (employee))))

g)

σ depend>2 (π Fname, Minit, Lname, depend (γ Ssn, Fname, Minit, Lname; Count(Ssn) -> depend (π Ssn, Fname, Minit, Lname (employee) ⨝ Ssn=Essn π Essn, Dependent\_name (dependent))))

h)

π Fname, Minit, Lname (π Mgr\_ssn (department) ⨝ Mgr\_ssn=Ssn σ Essn=null (π Ssn, Fname, Minit, Lname (employee) ⟕ Ssn=Essn π Essn, Dependent\_name (dependent)))

i)

X = π Fname, Minit, Lname, Address, Dno (σ Plocation='Aveiro' (employee ⨝ Ssn=Essn works\_on ⨝ Pno=Pnumber project))

Y = π Dnumber (σ Dlocation!='Aveiro' (employee ⨝ Ssn=Essn works\_on ⨝ Dno=Dnumber department ⨝ dept\_location))

π Fname, Minit, Lname, Address ( X ⨝ Dno=Dnumber Y)

5.2 a)

π nif, nome (σ encomenda.numero=null (fornecedor ⟕ nif=fornecedor encomenda))

b)

π codProd; avg(unidades) -> media\_unidades (item)

c)

π avg(num\_prod) -> avg\_prod\_enc (γ numEnc; sum(codProd) ->num\_prod item)

d)

?

5.3

a)

π numUtente, nome (σ numPresc=null (π numUtente, nome (paciente) ⟕ prescricao))

b)

π especialidade, num (γ especialidade; count(especialidade) -> num ((prescricao ⨝ numMedico=numSNS medico)))

c)

π farmacia, num (γ farmacia; count(numPresc) -> num (σ farmacia != null (prescricao)))

d) ?

e)

π farmacia, numRegFarm, num (γ farmacia, numRegFarm; count(prescricao.numPresc) -> num (σ farmacia!=null (π prescricao.numPresc, farmacia (prescricao)) ⨝ prescricao.numPresc=presc\_farmaco.numPresc (presc\_farmaco)))

f)

π numUtente (σ num>1 (π numUtente,num (γ numUtente; count(numUtente) -> num (π numUtente, numMedico, num (γ numUtente, numMedico; count(numUtente) -> num prescricao)))))