

# FBD

Natallia Kotsina

December 2021

## 1 Ejercicios sobre Algebra y Cálculo Relacional

1. Sobre las relaciones Realizar las siguientes consultas en Algebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- (a) Encontrar los códigos de los proveedores que suministran alguna pieza a J1.  
AR:

$$\pi_{codpro}(\sigma_{codpj=J1}(VENTAS))$$

CRT:

$$\{x.codpro | VENTAS(x) \wedge x.codpj = 'J1'\}$$

- (b) Encontrar los suministros cuya cantidad supere las 100 unidades.  
AR:

$$\pi_{codpro}(\sigma_{cantidad>100}(VENTAS))$$

CRT:

$$\{x.codpro | VENTAS(x) \wedge x.cantidad > 100\}$$

- (c) Encontrar los nombres de proveedores, piezas y proyectos que se encuentren en la misma ciudad.  
AR:

$$\pi_{nompro,nompie,nompj}(\sigma_{proveedor.ciudad=pieza.ciudad \wedge pieza.ciudad=proyecto.ciudad} (PROVEEDOR \times (PIEZA \times PROYECTO)))$$

CRT:

$$\{pr.nompro, pz.nompie, pj.nompj | PROVEEDOR(pr) \wedge PIEZA(pz) \wedge PROYECTO(pj) \wedge pr.ciudad = pz.ciudad \wedge pz.ciudad = pj.ciudad\}$$

- (d) Encontrar los nombres de las piezas suministradas por los proveedores de Londres.  
AR:

$$\pi_{PIEZAS.nompie}(PIEZAS \bowtie (VENTAS \bowtie \sigma_{ciudad=Londres}(PROVEEDOR)))$$

CRT:

$$\{pz.nompie | PIEZAS(pz) \wedge ((\exists v)(\exists pr)(VENTAS(v) \wedge PROVEEDORES(pr) \wedge v.codpro = pr.codpro \wedge v.codpie = pz.codpie \wedge pr.ciudad = 'Londres'))\}$$

- (e) Encontrar todas las parejas de ciudades tales que la primera sea la de un proveedor y la segunda la de un proyecto entre los cuales haya algun suministro.

AR:

$$\pi_{PROYECTO.ciudad, PROVEEDOR.ciudad}((VENTAS \bowtie PROYECTO) \bowtie PROVEEDOR)$$

CRT:

$$\{pr.ciudad, pj.ciudad | PROVEEDOR(pr) \wedge PROYECTO(pj) \wedge ((\exists v)(VENTAS(v) \wedge v.codpro = pr.codpro \wedge v.codpj = pj.codpj))\}$$

- (f) Encontrar los codigos de las piezas suministradas a algun proyecto por un proveedor que se encuentre en la misma ciudad que el proyecto.

AR:

$$\pi_{VENTAS.codpie}(\sigma_{proveedor.ciudad=proyecto.ciudad}((VENTAS \bowtie PROVEEDOR) \bowtie PROYECTO))$$

CRT:

$$\{v.codpie | VENTAS(v) \wedge ((\exists pj)(\exists pr)(PROYECTO(pj) \wedge PROVEEDOR(pr) \wedge v.codpj = pj.codpj \wedge v.codpro = pr.codpro \wedge pj.ciudad = pr.ciudad))\}$$

- (g) Encontrar los codigos de los proyectos que tienen al menos un proveedor que no se encuentre en su misma ciudad.

AR:

$$\pi_{VENTAS.codpie}(\sigma_{proveedor.ciudad \neq proyecto.ciudad}((VENTAS \bowtie PROVEEDOR) \bowtie PROYECTO))$$

CRT:

$$\{v.codpj | VENTAS(v) \wedge ((\exists pj)(\exists pr)(PROYECTO(pj) \wedge PROVEEDOR(pr) \wedge v.codpj = pj.codpj \wedge v.codpro = pr.codpro \wedge pr.ciudad \neq pj.ciudad))\}$$

- (h) Mostrar todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos.

AR:

$$\pi_{PIEZA.ciudad}(VENTAS \bowtie PIEZA) \cup \pi_{PROYECTO.ciudad}(VENTAS \bowtie PROYECTO)$$

CRT:

$$\{pz.ciudad, pj.ciudad | PIEZA(pz) \wedge PROYECTO(pj) \wedge (\exists v)(VENTAS(v) \wedge (v.codpie = pz.codpie \vee v.codpj = pz.codpj))\}$$

- (i) Mostrar todas las ciudades de los proveedores en las que no fabriquen piezas.

AR:

$$\pi_{ciudad}(PROVEEDOR) - \pi_{ciudad}(VENTAS \bowtie PIEZA)$$

CRT:

$$\{pr.ciudad | PROVEEDOR(pr) \wedge (\neg \exists pz)(\neg \exists v)(PIEZAS(pz) \wedge VENTAS(v) \wedge v.codpro = pr.codpro \wedge v.codpie = pz.codpie \wedge pr.ciudad = pz.ciudad)\}$$

- (j) Mostrar todas las ciudades de los proveedores en las que ademas se fabriquen piezas.  
AR:

$$\pi_{ciudad}(PROVEEDOR) \cap \pi_{ciudad}(VENTAS \bowtie PIEZA)$$

CRT:

$$prC(c) := \{pr.ciudad | PROVEEDOR(pr)\}$$

$$pzC(c) := \{pz.ciudad | PIEZA(pz)\}$$

$$\{x.ciudad | prC(x) \wedge pzC(x)\}$$

- (k) Encontrar los codigos de los proyectos que usan una pieza que vende S1.  
AR:

$$\pi_{codpj}(\sigma_{codpro=S1}(VENTAS))$$

CRT:

$$\{v.codpj | VENTAS(v) \wedge (v.codpro = 'S1')\}$$

- (l) Encontrar la cantidad mas pequena enviada en algun suministro.  
AR:

$$\rho(VENTAS) = V$$

$$\pi_{V.cantidad}(V) - \pi_{V.cantidad}(\sigma_{V.cantidad > VENTAS.cantidad}(V \times VENTAS))$$

CRT:

$$\{v.cantidad | VENTAS(v) \wedge (\neg \exists v2)(VENTAS(v2) \wedge v.cantidad > v2.cantidad)\}$$

- (m) Encontrar los codigos de los proyectos que no utilizan una pieza roja suministrada por un proveedor de Londres.  
AR:

$$\pi_{VENTAS.codpj}((VENTAS \bowtie \sigma_{ciudad=Londres}(PROVEEDOR)) \bowtie \sigma_{color \neq rojo}(PIEZA))$$

CRT:

$$\{v1.codpj | VENTAS(v) \wedge (\exists pr)(\exists pz)(PROVEEDOR(pr) \wedge PIEZA(pz)$$

$$\wedge v.codpro = pr.codpro \wedge v.codpie = pz.codpie \wedge$$

$$pz.color \neq 'rojo' \wedge pr.ciudad = 'Londres')\}$$

- (n) Encontrar los códigos de los proyectos que tienen como único proveedor a S1.  
AR:

$$\pi_{codpj}(VENTAS) - \pi_{codpj}(\sigma_{codpro \neq S1}(VENTAS))$$

CRT:

$$\{v1.codpj | VENTAS(v1) \wedge (\neg \exists v2)(VENTAS(v2) \wedge v2.codpro \neq S1' \wedge v1.codpj = v2.codpj)\}$$

- (o) Encontrar los codigos de las piezas que se suministran a todos los proyectos de Paris.  
AR:

$$\pi_{codpie, codpj}(VENTAS) \div \pi_{codpj}(\sigma_{ciudad=Paris}(PROYECTO))$$

CRT:

$$\{v.codpie | VENTAS(v) \wedge (\neg \exists pj)((PROYECTO(pj) \wedge pj.ciudad = 'Paris') \wedge (\neg \exists v2)(VENTAS(v2) \wedge pj.codpj = v2.codpj \wedge v2.codpie = v.codpie))\}$$

- (p) Encontrar los codigos de los proveedores que venden la misma pieza a todos los proyectos.  
AR:

$$\pi_{codpro}(\pi_{codpro, codpie, codpj}(VENTAS) \div \pi_{codpj}(PROYECTOS))$$

CRT:

$$\{v.codpro | VENTAS(v) \wedge (\neg \exists pj)((PROYECTO(pj)) \wedge (\neg \exists v2)(VENTAS(v2) \wedge v2.codpie = v.codpie \wedge v2.codpro = v.codpro \wedge v2.codpj = pj.codpj))\}$$

- (q) Encontrar los codigos de los proyectos a los que S1 suministra todas las piezas existentes.  
AR:

$$\pi_{codpj, codpie}(VENTAS) \div \pi_{codpie}(\sigma_{codpro=S1}(VENTAS))$$

CRT:

$$\{v1.codpj | VENTAS(v1) \wedge (\neg \exists v2)((VENTAS(v2) \wedge v2.codpro = 'S1') \wedge (\neg \exists v3)(VENTAS(v3) \wedge v3.codpie = v2.codpie \wedge v3.codpj = v1.codpj))\}$$

- (r) Mostrar los codigos de los proveedores que suministran todas las piezas a todos los proyectos.  
AR:

$$(\pi_{codpro, codpie, codpj}(VENTAS) \div \pi_{codpie}(PIEZA)) \div \pi_{codpj}(PROYECTO)$$

CRT:

Los codigos de los proveedores que suministran todas las piezas:

$$GRUPO(x) := \{v1.codpro, v1.codpj | VENTAS(v1) \wedge (\neg \exists v2)(VENTAS(v2) \wedge (\neg \exists v3)(VENTAS(v3) \wedge v3.codpie = v2.codpie \wedge v3.codpj = v1.codpj))\}$$

Los codigos de los proveedores que suministran todas las piezas a todos los proyectos:

$$\{x1.codpro | GRUPO(x1) \wedge (\neg \exists pj)(PROYECTO(pj) \wedge (\neg \exists x2)(GRUPO(x2) \wedge x2.codpj = pj.codpj \wedge x1.codpro = x2.codpro))\}$$

2. Realizar las siguientes consultas en Algebra Relacional y en Calculo Relacional Orientado a Tuplas:

- (a) Encontrar los regalos (descripcion) que no han sido reservados.

AR:

$$\pi_{LISTA\_BODA.ref\#,LISTA\_BODA.descripcion}(LISTA\_BODA) - \pi_{LISTA\_BODA.ref\#,LISTA\_BODA.descripcion}(RESERVA\_REGALO \bowtie LISTA\_BODA)$$

CRT:

$$\{L.descripcion | LISTA\_BODA(L) \wedge (\neg \exists R)(RESERVA\_REGALO(R) \wedge L.ref\# = R.ref\#)\}$$

- (b) Encontrar la direccion de los invitados que confirman la asistencia de mas de dos personas.

AR:

$$\pi_{direccion}(\sigma_{numero > 2}(INVITACIONES \bowtie CONFIRMAN))$$

CRT:

$$\{i.direccion | INVITACIONES(i) \wedge (\exists c)(CONFIRMAN(c) \wedge i.nombre = c.nombre \wedge c.numero > 2)\}$$

- (c) Encontrar el nombre y la referencia del regalo mas caro ya reservado.

AR:

$$\rho(LISTA\_BODA) = L$$

$$\pi_{RESERVA\_REGALO.ref\#,RESERVA\_REGALO.nombre}(RESERVA\_REGALO \bowtie (\pi_{LISTA\_BODA.ref\#}(LISTA\_BODA) - \pi_{LISTA\_BODA.ref\#}(\sigma_{LISTA\_BODA.precio < L.precio}(LISTA\_BODA \times L))))$$

CRT:

$$\{r.ref\#, r.nombre | RESERVA\_REGALO(r) \wedge (\exists L1)(LISTA\_BODA(L1) \wedge (\neg \exists L2)(LISTA\_BODA(L2) \wedge L1.precio < L2.precio \wedge r.ref\# = L1.ref\#))\}$$

3. Realizar las siguientes consultas en Algebra Relacional y en Calculo Relacional Orientado a Tuplas:

- (a) Hallar las parejas de hombres y mujeres que se caen mutuamente simpaticos, con edades entre 20 y 30 anos y que no esten casados entre si.

AR:

$$\pi_{NomH, NomM}(((\pi_{nomM, nomH}(\sigma_{HsimM.NomH = MsimH.NomH \wedge HsimM.NomM = MsimH.NomM} (HsimM \times MsimH))) \bowtie \sigma_{edad > 20 \wedge edad < 30}(HOMBRES)) \bowtie \sigma_{edad > 20 \wedge edad < 30}(MUJERES)) - \pi_{NomH, NomM}(MATRIM)$$

CRT:

- (b) Hallar las mujeres casadas a las que no cae simpático su marido.  
AR:

$$\pi_{nomM}(MATRIM) - \pi_{nomM}(\sigma_{MATRIM.nomH=HsimM.nomH \wedge MATRIM.NomM=HsimM.NomM}(MATRIM \times HsimM))$$

CRT:

$$\{m.nomM | MATRIM(m) \wedge (\neg \exists y)(HsimM(y) \wedge m.NomM = y.NomM \wedge m.NomH = y.NomH)\}$$

- (c) Hallar los hombres a los que no les cae simpática ninguna mujer.  
AR:

$$\pi_{nomH}(HOMBRES) - \pi_{NomH}(MsimH)$$

CRT:

$$\{h.NomH | HOMBRES(h) \wedge \neg MsimH(h)\}$$

- (d) Hallar las mujeres casadas a las que no les cae simpático ningún hombre casado.  
AR:

$$\pi_{nomM}(MATRIM) - \pi_{NomM}(\sigma_{HsimM.nomH=MATRIM.NomH}(HsimM \times MATRIM))$$

CRT:

$$\{m.NomM | MATRIM(m) \wedge (\neg \exists x)(HsimM(x) \wedge m.NomM = x.NomM)\}$$

4. Realizar las siguientes consultas en Álgebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- (a) Encontrar entre que dos ciudades se realiza el viaje más largo.  
AR:

$$\begin{aligned} \rho(RUTA) &= R \\ \pi_{ruta\#, ciudad\_sal, ciudad\_lleg}(RUTA) &- \\ \pi_{RUTA.ruta\#, RUTA.ciudad\_sal, RUTA.ciudad\_lleg}(\sigma_{RUTA.km < R.km}(RUTA \times R)) \end{aligned}$$

CRT:

$$\{r1.ruta\#, r1.ciudad\_sal, r1.ciudad\_lleg | RUTA(r1) \wedge (\neg \exists r2)(RUTA(r2) \wedge r1.km < r2.km)\}$$

- (b) Listar los nombres de los conductores que hayan llevado todos los camiones de la empresa.  
AR:

$$\pi_{nombre}(CONDUCTOR \bowtie (\pi_{DNI, matricula}(VIAJE) \div \pi_{matricula}(VEHICULO)))$$

CRT:

$$\begin{aligned} DNI(c) &:= \{v1.DNI | VIAJE(v1) \wedge (\neg \exists x)(VEHICULO(x) \\ &\wedge (\neg \exists v2)(VIAJE(v2) \wedge x.matricula = v2.matricula \wedge v2.viaje\# = v1.viaje\#))\} \end{aligned}$$

$$\{c.nombre | CONDUCTOR(c) \wedge (\exists x)(DNI(x) \wedge c.DNI = x.DNI)\}$$

- (c) Encontrar que días de la semana se hacen viajes entre Granada y Sevilla por la mañana (antes de las 13h.).

AR:

$$\pi_{dia\_sem}(\sigma_{hora\_sal < 13}(PROG\_VIAJE) \bowtie \sigma_{ciudad\_sal=Granada \wedge ciudad\_lleg=Sevilla}(RUTA))$$

CRT:

$$\{p.dia\_sem | (PROG\_VIAJE(p) \wedge p.hora\_sal < 13) \wedge (\exists r)(RUTA(r) \wedge p.ruta\# = r.ruta\# \wedge r.ciudad\_sal = 'Granada' \wedge r.ciudad\_lleg = 'Sevilla')\}$$

- (d) Encontrar las rutas que se hacen todos los días de la semana, suponiendo que hay viajes todos los días.

AR:

$$\pi_{ruta\#, dia\_sem}(PROG\_VIAJE) \div \pi_{dia\_sem}(PROG\_VIAJE)$$

CRT:

$$\{p1.ruta\# | PROG\_VIAJE(p1) \wedge (\neg \exists p2)(PROG\_VIAJE(p2) \wedge (\neg \exists p3)(PROG\_VIAJE(p3) \wedge p3.dia\_sem = p2.dia\_sem \wedge p3.ruta\# = p1.ruta\#))\}$$

5. Realizar las siguientes consultas en Algebra Relacional y en Calculo Relacional Orientado a Tuplas:

- (a) Listar las provincias que son visitadas por todos los representantes.

AR:

$$\pi_{provincia, DNI}(ZONA\_REP) \div \pi_{DNI}(REPRESENTANTE)$$

CRT:

$$\{z1.provincia | ZONA\_REP(z1) \wedge (\neg \exists r)(REPRESENTANTE(r) \wedge (\neg \exists z2)(ZONA\_REP(z2) \wedge r.DNI = z2.DNI \wedge z1.provincia = z2.provincia))\}$$

- (b) Encontrar los representantes que venden fuera de su provincia artículos fabricados en su provincia.

AR:

$$\pi_{DNI}(\sigma_{REPRESENTANTE.provincia <> ZONA\_REP.provincia} (REPRESENTANTE \bowtie ZONA\_REP)) - \pi_{DNI}(\sigma_{REPRESENTANTE.provincia <> ARTICULO.prov\_fab} ((REPRESENTANTE \bowtie PEDIDOS) \bowtie ARTICULO))$$

CRT:

$$repUNO(r) := \{r.DNI | REPRESENTANTE(r) \wedge (\exists z)(ZONA\_REP(z) \wedge r.provincia \neq z.provincia)\}$$

$$repDOS(r) := \{r.DNI | REPRESENTANTE(r) \wedge PEDIDOS(p) \wedge ARTICULO(a) \wedge r.DNI = p.DNI \wedge p.cod\_art = a.cod\_art \wedge r.provincia = a.prov\_fab\}$$

$$\{r1.DNI | repUNO(r1) \wedge (\neg \exists r2)(repDOS(r2) \wedge r1.DNI = r2.DNI)\}$$

- (c) Obtener las poblaciones de Granada que hayan superado los 50.000 euros de facturación y quien realizó el pedido.

AR:

$$\pi_{ZONA\_REP.poblacion}(\sigma_{ZONA\_REP.poblacion=PEDIDOS.poblacion}(\sigma_{provincia=Granada}(ZONA\_REP) \times \sigma_{cantidad>50000}(PEDIDOS)))$$

CRT:

$$\{z.poblacion|ZONA\_REP(z) \wedge PEDIDOS(p) \wedge (z.provincia = 'Granada' \wedge p.cantidad > 50000 \wedge z.poblacion = p.poblacion)\}$$

- (d) Mostrar las zonas que incluyen a una sola población.

AR:

$$\rho(ZONA\_REP) = Z$$

$$\pi_{ZONA\_REP.cod\_zona}(ZONA\_REP) - \pi_{ZONA\_REP.cod\_zona}(\sigma_{ZONA\_REP.cod\_zona=Z.cod\_zona \wedge ZONA\_REP.poblacion <> Z.poblacion}(ZONA\_REP \times Z))$$

CRT:

$$\{z1.cod\_zona|ZONA\_REP(z1) \wedge (\neg \exists z2)(ZONA\_REP(z2) \wedge z1.cod\_zona = z2.cod\_zona \wedge z1.poblacion \neq z2.poblacion)\}$$

- (e) Encontrar el código del artículo vendido en mayor cantidad.

AR:

$$\rho(PEDIDOS) = P$$

$$\pi_{PEDIDOS.cod\_art}(PEDIDOS) - \pi_{PEDIDOS.cod\_art}(\sigma_{PEDIDOS.cantidad < P.cantidad}(PEDIDOS \times P))$$

CRT:

$$\{p1.cod\_art|PEDIDOS(p1) \wedge (\neg \exists p2)(PEDIDOS(p2) \wedge p1.cod\_art \neq p2.cod\_art \wedge p1.cantidad < p2.cantidad)\}$$

6. Teniendo en cuenta que tanto los coordinadores como los ponentes han de estar inscritos en el congreso, resolver las siguientes consultas mediante Álgebra Relacional y mediante Cálculo Relacional Orientado a Tuplas:

- (a) Mostrar los nombres de los ponentes que coordinan su propia sesión.

AR:

$$\pi_{INSCRIPCIONES.nombre}(\sigma_{INSCRIPCIONES.ins\#=PROGRAMA.ponente\#}((\sigma_{PROGRAMA.ponente\#=SESIONES.coord\#}(PROGRAMA \bowtie SESIONES)) \times INSCRIPCIONES))$$

CRT:

$$\{p.ponente\#|PROGRAMA(p) \wedge (\exists s)(SESIONES(s) \wedge p.ses\# = s.ses\# \wedge p.ponente\# = s.coord\#)\}$$



- (b) Seleccionar los coordinadores que coordinan una unica sesion.  
AR:

$$\rho(SESSIONES) = S$$

$$\pi_{SESSIONES.coord}(SESSIONES) - \pi_{SESSIONES.coord} \\ (\sigma_{SESSIONES.ses\# = S.ses\# \wedge SESSIONES.coord < > S.coord}(SESSIONES \times S))$$

CRT:

$$\{s1.coord\# | SESSIONES(s1) \wedge (\neg \exists s2) \\ (SESSIONES(s2) \wedge (s1.ses\# \neq s2.ses\#) \wedge (s1.coord\# = s2.coord\#))\}$$

- (c) Mostrar el titulo de los articulos que se exponen en primer y ultimo lugar.  
AR:

$$\rho(PROGRAMA) = P$$

$$\pi_{titulo}(ARTICULOS \bowtie \pi_{PROGRAMA.art\#,PROGRAMA.hora_inicio}(PROGRAMA) - \\ \pi_{PROGRAMA.art\#,PROGRAMA.hora_inicio} \\ (\sigma_{PROGRAMA.hora_inicio > P.hora_inicio}(PROGRAMA \times P))) \\ \cup \\ \pi_{titulo}(ARTICULOS \bowtie \pi_{PROGRAMA.art\#,PROGRAMA.hora_fin}(PROGRAMA) - \\ \pi_{PROGRAMA.art\#,PROGRAMA.hora_fin} \\ (\sigma_{PROGRAMA.hora_fin < P.hora_fin}(PROGRAMA \times P)))$$

CRT:

El titulo de los articulos que se exponen en primer lugar:

$$inTITULO(t) := \{a.titulo | ARTICULO(a) \wedge PROGRAMA(p1) \wedge a.art\# = p1.art\# \\ \wedge (\neq \exists p2)(PROGRAMA(p2) \wedge (p1.hora_inicio < p2.hora_inicio))\}$$

El titulo de los articulos que se exponen en ultimo lugar:

$$finTITULO(t) := \{a.titulo | ARTICULO(a) \wedge PROGRAMA(p1) \wedge a.art\# = p1.art\# \\ \wedge (\neq \exists p2)(PROGRAMA(p2) \wedge (p1.hora_fin > p2.hora_fin))\}$$

El titulo de los articulos que se exponen en primer y ultimo lugar:

$$\{t.titulo | inTITULO(t) \vee finTITULO(t)\}$$