

AlgebraCalculoRelacional.pdf



martasw99



Fundamentos de Bases de Datos



3º Doble Grado en Ingeniería Informática y Matemáticas

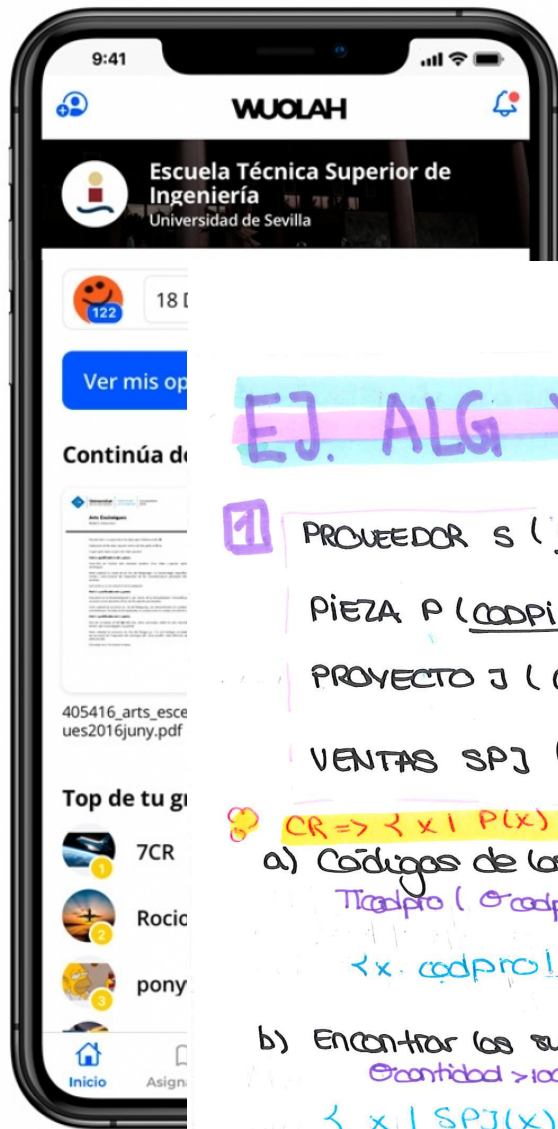


**Facultad de Ciencias
Universidad de Granada**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



EJ. ALG Y CÁLCULO RELACIONAL

Algebra

Cálculo

1 PROVEEDOR S (CODPRO, NOMPRO, STATUS, CIUDAD)

PIEZA P (CODPIE, NOMPIE, COLOR, PESO, CIUDAD)

PROYECTO J (CODPJ, NOMPJ, CIUDAD)

VENTAS SPJ (CODPRO, CODPIE, CODPJ, CANTIDAD, FECHA)

CR $\Rightarrow \{ x \mid P(x) \wedge \exists y, z (SPJ(y) \wedge \dots \wedge PC(z) \wedge \dots) \}$

a) Códigos de los proveedores que suministran alguna pieza a J1
 $\Pi_{\text{codpro}} (\sigma_{\text{codpj} = J1} (SPJ))$

$\{ x.\text{codpro} \mid S(x) \wedge \exists y (SPJ(y) \wedge y.\text{codpro} = x.\text{codpro} \wedge$
 $\wedge y.\text{codpj} = J1) \}$ // $\{ x.\text{codpro} \mid SPJ(x) \wedge x.\text{codpj} = J1 \}$

b) Encontrar los suministros cuya cantidad > 100
 $\sigma_{\text{cantidad} > 100} (SPJ)$

$\{ x \mid SPJ(x) \wedge x.\text{cantidad} > 100 \}$

c) nombres de proveedores, piezas y proyectos que se encuentren en
== ciudad

$\Pi_{\text{codpro}, \text{codpj}, \text{codpie}} ((S \bowtie_{\text{ciudad}} P) \bowtie_{\text{ciudad}} J)$

$\{ x.\text{nompro}, y.\text{nompie}, z.\text{nompj} \mid S(x) \wedge P(y) \wedge J(z) \wedge x.\text{ciudad} = y.\text{ciudad}$
 $\wedge y.\text{ciudad} = z.\text{ciudad} \}$

d) nombres de las piezas suministradas por los proveedores de

Londres $\Pi_{\text{nom}} (\sigma_{\text{p.codpie} = (P \times (\sigma_{\text{spj.codpro} = (SPJ \times \sigma_{\text{ciudad} = \text{Londres}} (S)))) } (SPJ \bowtie_{\text{ciudad}} S))$

$\{ x.\text{nompie} \mid P(x) \wedge \exists y, z (SPJ(y) \wedge S(z) \wedge y.\text{codpie} = x.\text{codpie} \wedge$
 $\wedge y.\text{codpro} = z.\text{codpro} \wedge z.\text{ciudad} = \text{Londres}) \}$

e) Encontrar todas las parejas de ciudades tales que la primera
sea la de un proveedor y la 2ª de un proyecto entre los cuales
haya algún suministro

$\Pi_{\text{s.ciudad}, \text{j.ciudad}} (\sigma_{\text{spj.codpj} = \text{j.codpj}} (\sigma_{\text{spj.codpro} = \text{s.codpro}} (SPJ \times S)) \times J)$

$\{ x.\text{ciudad}, y.\text{ciudad} \mid S(x) \wedge J(y) \wedge \exists z (SPJ(z) \wedge$
 $\wedge z.\text{codpro} = x.\text{codpro} \wedge z.\text{codpj} = y.\text{codpj}) \}$

f) los cod. de las piezas suministradas a algún proyecto
por un proveedor que se encuentre en = ciudad que proyecto

$A = \Pi_{\text{codpro}, \text{codpj}} (\sigma_{\text{s.ciudad} = \text{j.ciudad}} (S \times J))$

$\Pi_{\text{codpie}} (SPJ \bowtie A)$

$\{ x.\text{codpie} \mid P(x) \wedge \exists y, z, t (SPJ(y) \wedge S(z) \wedge J(t) \wedge y.\text{codpie} = x.\text{codpie}$
 $\wedge y.\text{codpro} = z.\text{codpro} \wedge y.\text{codpj} = t.\text{codpj} \wedge z.\text{ciudad} = t.\text{ciudad}) \}$

g) los códigos de los proyectos que tienen al menos un proveedor que no se encuentre en su misma ciudad.

$$\pi_{\text{codpj}} (\sigma_{s.\text{ciudad} \neq p.\text{ciudad}} ((SPJ \bowtie S) \bowtie J))$$

$$\{x.\text{codpj} \mid \exists x \wedge \exists y, z (SPJ(y) \wedge S(z) \wedge y.\text{codpro} = z.\text{codpro} \wedge y.\text{codpj} = x.\text{codpj} \wedge \neg (x.\text{ciudad} = z.\text{ciudad}))\}$$

h) Todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos

$$\pi_{\text{ciudad}} (P) \cup \pi_{\text{ciudad}} (J)$$

$$\{x.\text{ciudad} \mid P(x) \cup J(x)\}$$

i) Todas las ciudades de los proveedores en las que no fabriquen piezas

$$\pi_{\text{ciudad}} (S) - \pi_{\text{ciudad}} (P)$$

$$\{x.\text{ciudad} \mid S(x) \wedge \neg (\exists y (P(y) \wedge y.\text{ciudad} = x.\text{ciudad}))\}$$

j) Todas las ciudades de los proveedores en las que además se fabriquen piezas

$$\pi_{\text{ciudad}} (S) \cap \pi_{\text{ciudad}} (P)$$

$$\{x.\text{ciudad} \mid S(x) \wedge \exists y (P(y) \wedge y.\text{ciudad} = x.\text{ciudad})\}$$

k) Códigos de proyectos que usan una pieza que vende S1

$$\pi_{\text{codpj}} (SPJ \bowtie \pi_{\text{codpie}} (\sigma_{\text{codpro} = S1} (SPJ)))$$

$$\{x.\text{codpj} \mid SPJ(x) \wedge x.\text{codpro} = S1\}$$

l) Encontrar la cantidad más pequeña enviada en algún suministr.

$$A = \pi_{\text{cantidad}} (SPJ) \quad B = A \quad C = \pi_{A.\text{cantidad}} (\sigma_{A.\text{cant} > B.\text{cant}} (A \bowtie B)) \quad [A - C]$$

$$\{x.\text{cantidad} \mid SPJ(x) \wedge \neg (\exists y (SPJ(y) \wedge y.\text{cantidad} < x.\text{cantidad}))\}$$

x también estaría

m) los códigos de los proyectos que no utilizan una pieza roja suministrada por un proveedor de Londres

$$A = \pi_{\text{codpro}} (\sigma_{\text{ciudad} = \text{Londres}} (S)) \quad B = \pi_{\text{codpie}} (\sigma_{\text{color} = \text{"rojo"}} (P))$$

$$C = \pi_{\text{codpj}} ((SPJ \bowtie A) \bowtie B) \quad [\pi_{\text{codpj}} (J) - C]$$

$$\{x.\text{codpj} \mid J(x) \wedge \exists y, z, t (SPJ(x) \wedge P(z) \wedge S(t) \wedge x.\text{codpj} = y.\text{codpj} \wedge y.\text{codpie} = z.\text{codpie} \wedge y.\text{codpro} = t.\text{codpro} \wedge z.\text{color} = \text{rojo} \wedge t.\text{ciudad} = \text{Londres})\}$$



**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

n) los códigos de los proyectos que tienen como único proveedor a S1. $\pi_{\text{codpj}}(\sigma_{\text{codpro} \neq S1}(\text{SPJ}))$

$$\langle x.\text{codpj} \mid \text{SPJ}(x) \wedge \forall y(\text{SPJ}(y) \wedge x.\text{codpj} = y.\text{codpj}) \rightarrow y.\text{codpro} = S1 \rangle$$

$$\langle x.\text{codpj} \mid \text{SPJ}(x) \wedge \neg(\exists y(\text{SPJ}(y) \wedge x.\text{codpj} = y.\text{codpj} \wedge y.\text{codpro} \neq S1)) \rangle$$

ñ) los códigos de las piezas que se suministran a todos los proyectos de París $A = \pi_{\text{codpie, codpj}}(\text{SPJ}) \quad B = \pi_{\text{codpj}}(\sigma_{\text{codpro} = \text{Paris}}(J))$

$[A \div B]$ Candidato

Divisor

$$\langle x.\text{codpie} \mid P(x) \wedge \forall y(J(y) \wedge y.\text{codpro} = \text{Paris}) \rightarrow$$

Relación
Candidato - Divisor $\rightarrow \exists z(\text{SPJ}(z) \wedge z.\text{codpie} = x.\text{codpie} \wedge y.\text{codpj} = z.\text{codpj}) \rangle$

o) Códigos de los proveedores que venden la misma pieza a todos los proyectos. $A = \pi_{\text{codpie, codpj, codpro}}(\text{SPJ}) \div \pi_{\text{codpj}}(J)$

A: Pares proveedor - pieza que están en todos los proyectos $\Rightarrow \pi_{\text{codpro}}(A)$

$$\langle x.\text{codpro} \mid S(x) \wedge \exists y(P(y) \wedge \forall z(J(z) \rightarrow (\exists w(\text{SPJ}(w) \wedge w.\text{codpro} = x.\text{codpro} \wedge y.\text{codpie} = w.\text{codpie} \wedge z.\text{codpj} = w.\text{codpj})))) \rangle$$

p) los códigos de los proyectos a los que S1 suministra todas las piezas existentes \rightarrow Pares proveedor - proyecto que tienen todas las piezas

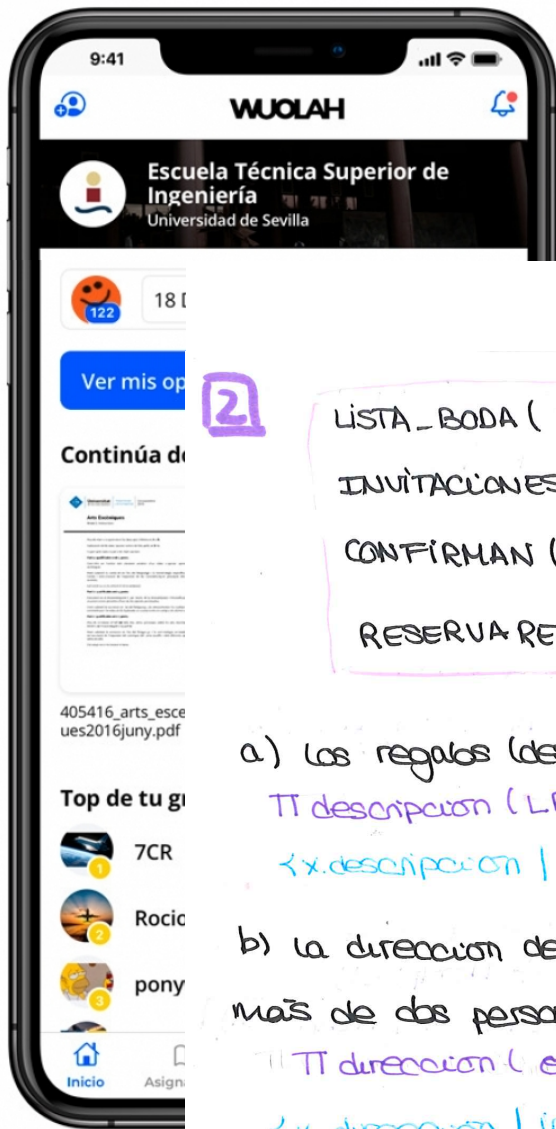
$$A = \pi_{\text{codpro, codpj, codpie}}(\text{SPJ}) \div \pi_{\text{codpie}}(P) \quad \pi_{\text{codpj}}(\sigma_{\text{codpro} = S1}(A))$$

$$\langle x.\text{codpj} \mid J(x) \wedge \forall y(P(y) \rightarrow \exists z(\text{SPJ}(z) \wedge z.\text{codpro} = S1 \wedge z.\text{codpj} = x.\text{codpj} \wedge z.\text{codpie} = y.\text{codpie})) \rangle$$

q) Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.

$$(\pi_{\text{codpie, codpj, codpro}}(\text{SPJ}) \div \pi_{\text{codpie}}(P)) \div \pi_{\text{codpj}}(J)$$

$$\langle x.\text{codpro} \mid S(x) \wedge \forall y, z((P(y) \wedge J(z)) \rightarrow \exists w(\text{SPJ}(w) \wedge w.\text{codpro} = x.\text{codpro} \wedge w.\text{codpie} = y.\text{codpie} \wedge w.\text{codpj} = z.\text{codpj})) \rangle$$



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



2

LISTA_BODA (REFx, DESCRIPCIÓN, PRECIO) LB
INVITACIONES (NOMBRE, DIRECCIÓN, CIUDAD)
CONFIRMAN (NOMBRE, NÚMERO)
RESERVA REGALO (NOMBRE, REFx, FECHA)

a) los regalos (descripción) que no han sido reservados

$TI_{descripción} (LB) - TI_{descripción} (LB \times RESERVA_REG)$
 $\{x.descripción \mid LIST_BODA(x) \wedge \neg(\exists y (RESERV(y) \wedge y.ref = x.ref))\}$

b) la dirección de los invitados que confirman la asistencia de más de dos personas

$TI_{dirección} (\exists num \geq 2. (CONFIRMAN) \times INVITACIONES)$
 $\{x.dirección \mid INVIT(x) \wedge \exists y (CONF(y) \wedge y.nombre = x.nombre \wedge y.número > 2)\}$

c) Encontrar el nombre y referencia del regalo **más caro** ya reservado

$A = TI_{REF, NOMBRE, PRECIO} (LISTA_BODA \times RESERVA_REGALO)$

$A = B$

$C = TI_{B.ref, B.nombre, B.precio} (\exists A.precio > B.precio (A \times B))$

A - C

$\{x.nombre, x.ref \mid RESERV(x) \wedge \exists y (LISTA_BODA(y) \wedge y.ref = x.ref) \wedge \nexists z ((LISTA_BODA(z) \wedge \exists w (RESERV(w) \wedge w.ref = z.ref)) \rightarrow z.precio \leq y.precio)\}$

3

HOMBRES (NomH, Edad)MUJERES (NomM, Edad)HSimM (NomH, NomM) El hombre H cae simpático a la mujer MMsimH (NomM, NomH) La mujer M " " al hombre HMatrim (NomH, NomM) H y M están casados

- a) Las parejas de hombres y mujeres que se caen mutuamente simpáticos con edades entre 20 y 30 y que no estén casados entre sí.

$$(HSimM \cap MsimH) \cap \left(\prod_{\text{NomH}} \left(\bigoplus_{\substack{20 \leq H.edad \leq 30 \\ 20 \leq M.edad \leq 30}} (H \times M) \right) - Matrim \right)$$

$$\{x.nomH, y.nomM \mid H(x) \wedge M(y) \wedge 20 \leq x.edad \leq 30 \wedge 20 \leq y.edad \leq 30 \wedge \\ \wedge \exists z, t (HSimM(z) \wedge MsimH(t) \wedge z.nomH = x.nomH \wedge z.nomM = y.nomM \wedge \\ t.nomM = y.nomM \wedge t.nomH = x.nomH \wedge \neg \exists m (Matrim(m) \wedge m.nomH = x.nomH \wedge \\ m.nomM = y.nomM))\}$$

- b) Hallar las mujeres casadas a las que no cae simpático su marido

$$\prod_{\text{NomM}} (Matrim - HsimM)$$

$$\{x.nomM \mid Matrim(x) \wedge \neg (\exists y (HSimM(y) \wedge y.nomM = x.nomM \wedge \\ \wedge y.nomH = x.nomH))\}$$

- c) Los hombres a los que no cae simpática ninguna mujer

$$\prod_{\text{NomH}} (Hombres) - \prod_{\text{NomH}} (HSimM)$$

$$\{x.nomH \mid H(x) \wedge \neg (\exists y (HSimM(y) \wedge y.nomH = x.nomH))\}$$

- d) Las mujeres casadas a las que no les cae simpático ningún hombre casado.

$$A = \prod_{\text{NomH}} (Matrim) \quad B = A \times HsimM \rightarrow \text{Hombres casados que les caen simp. a alguna mujer}$$

$$C = \prod_{\text{NomM}} (B) \rightarrow \text{Mujeres a las que les cae sim algún H casado}$$

$$\prod_{\text{NomM}} (Matrim) - C$$

$$\{x.nomM \mid Matrim(x) \wedge \neg (\exists y, z (HSimM(y) \wedge Matrim(z) \wedge \\ y.nomM = x.nomM \wedge y.nomH = z.nomH))\}$$

4

CONDUCTOR (DNI, NOMBRE, DIR, PROV)

VEHICULO (MATRÍCULA, CARGA-MAX, FECHA-COMPR)

RUTA (RUTA#, CIUDAD-SAL, CIUDAD-UEG, KM)

VIAJE (VIAJE#, DNI, MATRÍCULA)

PROG-VIAJE (VIAJE#, RUTA#, DÍA-SEM, HORA-SAL, HORA-UEG)

a) Entre qué dos ciudades se realiza el viaje más largo

$$A = \Pi_{ciudad-sal, ciudad-ueg, km} (Ruta) \quad [A - C]$$

$$A = B$$

$$C = \Pi_{ciudad-sal, ciudad-ueg, km} (\Theta_{A.km > B.km} (A \times B))$$

$$\{x.ciudad-sal, x.ciudad-ueg \mid Ruta(x) \wedge \forall y (Ruta(y) \rightarrow y.km \leq x.km)\}$$

b) Nombres de los conductores que hayan llevado todos los camiones de la empresa

$$\Pi_{nombre} (\Pi_{DNI, nombre, matricula} (VIAJE \bowtie CONDUCTOR)) \div \Pi_{matric} (VEHIC)$$

$$\{x.nombre \mid conductor(x) \wedge \forall y (Vehiculo(y) \rightarrow \exists z (VIAJE(z) \wedge z.dni = x.dni \wedge z.matricula = y.matricula))\}$$

c) Qué días de la semana se hacen viajes entre GRANADA y SEVILLA por la mañana (antes de las 13h)

$$\Pi_{dia-sem} (\Theta_{ciudad-sal=GR \wedge ciudad-ueg=SEV} (Ruta) \bowtie \Theta_{h-sal \leq 13} (Prog-viaje) \vee (ciudad-sal=SEV \wedge ciudad-ueg=GR))$$

$$\{x.dia \mid Prog-viaje(x) \wedge \exists y (Ruta(y) \wedge ((y.ciudad-sal=GR \wedge y.ciudad-ueg=SEV) \vee (y.ciudad-sal=SEV \wedge y.ciudad-ueg=GR)) \wedge y.ruta = x.ruta) \wedge x.hora-sal < 13\}$$

d) Las rutas que se hacen todos los días de la semana, suponiendo que hay viajes todos los días.

$$\Pi_{RUTA} (\Theta_{dia-sem=lunes} (Prog-viaje)) \cap \dots \cap \Pi_{RUTA} (\Theta_{dia-sem=domingo} (Prog-viaje))$$

$$\Pi_{RUTA, dia-sem} (Prog-viaje) \div \Pi_{dia-sem} (Prog-viaje)$$

$$\{x.ruta \mid Prog-viaje(x) \wedge \forall y (Prog-viaje(y) \rightarrow \exists z (Prog-viaje(z) \wedge y.dia-sem = z.dia-sem \wedge x.dia-sem = z.dia-sem))\}$$