

# Ejercicios de algebra relacional

## EJERCICIO 1

Sea la siguiente BD:

**PROVEEDORES**( nro-p,nom-p,categoría,ciud-p)

**ITEMS**(nro-i,descripción-i,ciud-i)

**PEDIDOS**(nro-p,nro-c,nro-i,cantidad,precio)

**CLIENTES**(nro-c,nom-c,ciud-c)

### 1. Listar los proveedores de Cordoba.

$\pi$  nom-p ( $\sigma$  ciud-p = 'cordoba' (PROVEEDORES))

SELECT NOM-P FROM PROVEEDORES

WHERE CIUD-P = 'CORDOBA';

### 2. Listar los proveedores que proveen el item "i1".

$\pi$  PROVEEDORES.nom-p, PROVEEDORES.nro-p  $\sigma$  descr-i = 'i1' PROVEEDORES  $\bowtie$  PROVEEDORES.nro-p = PEDIDOS.nro-p PEDIDOS  $\bowtie$  ITEMS.nro-i = PEDIDOS.nro-i ITEMS

$\pi$  PROVEEDORES.nom-p ( $\sigma$  ITEMS.descr-i = 'i1'((PROVEEDORES  $\bowtie$  PEDIDOS)  $\bowtie$  ITEMS))

SELECT P.Nro-P, P.Nom-P FROM ITEMS AS I, PEDIDOS AS PD, PROVEEDORES AS P WHERE I.Descripcion-I = 'I1' AND PD.Nro-I = I.Nro-I AND P.Nro-P = PD.Nro-P;

### 3. Listar los clientes que solicitan items provistos por "p1".

$\pi$  CLIENTES.nom-c  $\sigma$  PROVEEDORES.nom-p = 'p1' CLIENTES  $\bowtie$  CLIENTES.nro-c = PEDIDOS.nro-c PEDIDOS  $\bowtie$  PEDIDOS.nro-p = PROVEEDORES.nro-p PROVEEDORES

$\pi$  CLIENTES.nom-c ( $\sigma$  PROVEEDORES.nom-p = 'p1'((CLIENTES  $\bowtie$  PEDIDOS)  $\bowtie$  PROVEEDORES))

select CLIENTES.nom-c from CLIENTES inner join PEDIDOS on CLIENTES.nro-c = PEDIDOS.nro-c inner join PROVEEDORES on PEDIDOS.nro-p = PROVEEDORES.nro-p where PROVEEDORES.nom-p = 'p1';

#### **4. Listar los clientes que solicitan algún ítem provisto por proveedores con categoría mayor que 4.**

$\pi$  CLIENTES.nom-c  $\sigma$  PROVEEDORES.categoria > 4 CLIENTES  $\bowtie$  CLIENTES.nro-c = PEDIDOS.nro-c PEDIDOS  $\bowtie$  PEDIDOS.nro-p = PROVEEDORES.nro-p PROVEEDORES

$\pi$  CLIENTES.nom-c ( $\sigma$  PROVEEDORES.categoria < 4((CLIENTES  $\bowtie$  PEDIDOS)  $\bowtie$  PROVEEDORES))

select CLIENTES.nom-c from CLIENTES inner join PEDIDOS on CLIENTES.nro-c = PEDIDOS.nro-c inner join PROVEEDORES on PEDIDOS.nro-p = PROVEEDORES.nro-p where PROVEEDORES.categoria > 4;

#### **5. Listar los ítems pedidos por clientes de Rosario.**

$\pi$  ITEMS.descr-i  $\sigma$  CLIENTES.ciud-c = 'rosario' ITEMS  $\bowtie$  ITEMS.nro-i = PEDIDOS.nro-i PEDIDOS  $\bowtie$  PEDIDOS.nro-c = CLIENTES.nro-c CLIENTES

$\pi$  ITEMS.descr-i ( $\sigma$  CLIENTES.ciud-c = 'rosario'((ITEMS  $\bowtie$  CLIENTES)  $\bowtie$  PEDIDOS))

select ITEMS.descr\_i from ITEMS inner join PEDIDOS on ITEMS.nro\_i = PEDIDOS.nro\_i inner join CLIENTES on PEDIDOS.nro\_c = CLIENTES.nro\_c where CLIENTES.ciud\_c = 'rosario';

#### **6. Listar los pedidos en los cuales un cliente de Rosario solicita artículos fabricados en Mendoza (ciud-i = "Mendoza").**

$\pi$  PEDIDOS.cantidad ( $\sigma$  PROVEEDORES.nom-p = 'rosario'  $\sigma$  ITEMS.ciud-i = 'mendoza'(PEDIDOS  $\bowtie$  PROVEEDORES  $\bowtie$  ITEMS  $\bowtie$  CLIENTES))

$\pi$  cantidad (PEDIDOS)\*  $\pi$  nom-p = 'Rosario' (PROVEEDORES)\*  $\sigma$  ciud -i = 'Mendoza'(ÍTEMS).

SELECT PR.nom-p, CL.nom-c, IT.descripcion-i, PE.cantidad, PE.precio FROM proveedores AS PR, clientes AS CL, items AS IT, pedidos AS PE WHERE PR.nro-p =

PE.nro-p AND CL.nro-c = PE.nro-c AND IT.nro-i = PE.nro-i AND CL.ciud-c = 'rosario' AND IT.ciud-i = 'mendoza';

**7. Listar los pedidos en los que el cliente "23" solicita items no solicitados por el cliente "30".**

$\pi$  PEDIDOS.nro-p(PEDIDOS)  $\bowtie$  CLIENTES CLIENTES.nro-c = PEDIDOS.nro-i  $\bowtie$  ITEMS ITEMS..nro-i= PEDIDOS.nro-i ( $\sigma$ nro-c= 23(CLIENTES))  $\sim \pi$  ITEMS.nro-i(ITEMS)  $\bowtie$  ITEMS ITEMS..nro-i= PEDIDOS.nro-i ( $\sigma$ nro-c= 30(CLIENTES))

$\pi$  cantidad ( $\sigma$  nro-c =23) (PEDIDOS)  $\pi$  cantidad ( $\pi$  nro-c=30) (PEDIDOS)

SELECT PEDIDOS.nro-p FROM PEDIDOS INNER JOIN CLIENTES ON PEDIDOS.nro-c = CLIENTES.nro-c INNER JOIN ITEMS ON PEDIDOS.nro-i = ITEMS.nro-i WHERE CLIENTES.nro-c= 23 AND ITEMS.nro-i NOT IN (select ITEMS.nro-i FROM PEDIDOS INNER JOIN CLIENTES ON PEDIDOS.nro-c = CLIENTES.nro-c INNER JOIN ITEMS ON PEDIDOS.nro-i = ITEMS.nro-i WHERE CLIENTES.nro-c= 30)

**8. Listar las ciudades en la forma (ciu1,ciu2 ) tales que un proveedor en ciu1 provea items solicitados por clientes de ciu2.**

$\sigma$  Ciud-p=ciu1 ( $\pi$  ciud-c=ciu2)(PROVEEDORES)(CLIENTES)

SELECT DISTINCT P.Ciud-P AS Ciudad1, C.Ciud-C AS Ciudad2 FROM PROVEEDORES AS P, CLIENTES AS C, PEDIDOS AS PD WHERE P.Nro-P = PD.Nro-P AND C.Nro-C = PD.Nro-C;

**9. Listar los números de proveedores cuya categoría sea mayor que la de todos los proveedores que proveen el item "cuaderno".**

$\pi$  nro-p ( $\sigma$  categoría> Todos proveedores) ( $\pi$  descripción-i =cuaderno)(PROVEEDORES)( ítems)

SELECT P.Nro-P, P.Nom-P FROM PROVEEDORES AS P WHERE P.Categoria > ( SELECT MAX(P.Categoria) FROM PROVEEDORES AS P, PEDIDOS as PD, ITEMS AS I WHERE I.Descripcion-I = 'Cuaderno' AND PD.Nro-I = I.Nro-I AND P.Nro-P = PD.Nro-P;

**10. Listar los clientes que han pedido 2 o más ítems distintos.**

$\pi \text{ nom-c } (\sigma \text{ cantidad} > 2) * \pi \text{ nro-i} > 2 \text{ (PEDIDOS)}$

SELECT DISTINCT PD1.Nro-C, C.Nom-C FROM PEDIDOS AS PD1, PEDIDOS AS PD2, CLIENTES AS C WHERE PD1.Nro-C = PD2.Nro-C AND PD1.Nro-I <> PD2.Nro-I AND C.Nro-C = PD1.Nro-C;

**11. Listar los proveedores que proveen a todos los clientes de Córdoba una cantidad mayor que el promedio de las cantidades pedidas por los clientes de Rosario.**

$\pi \text{ nom-p } (\sigma \text{ ciud-p=córdoba}) * \pi \text{ cantidad} > \Sigma \text{ cantidades } (\sigma \text{ nom-p=N.de cantidad Rosario}) \text{ (PROVEEDORES, PEDIDOS)}$

SELECT P.Nro-P, P.Nom-P FROM PROVEEDORES AS P, PEDIDOS AS PD, CLIENTES AS C WHERE C.Ciud-C = 'Cordoba' AND PD.Nro-C = C.Nro-C AND PD.Cantidad > ( SELECT AVG(SUMA) FROM TEMP ) AND P.Nro-P = PD.Nro-P GROUP BY P.Nro-P, P.Nom-P HAVING COUNT(C.Nro-C) = ( SELECT COUNT(C2.Nro-C) FROM CLIENTES C2 , PEDIDOS PD2 WHERE C2.Ciud-C = 'Cordoba' AND C2.Nro-C = PD2.Nro-C AND PD2.Nro-P = P.Nro-P );

**EJERCICIO 2**

Sea la siguiente Base de Datos:

**VUELOS** (nro-vuelo, desde, hasta)

**AVION-UTILIZADO** (nro-vuelo, tipo-avión, nro-avión)

**INFO-PASAJEROS** (nro-vuelo, dni, nombre, origen, destino)

Los vuelos no pueden tener más de dos escalas y no hay cambio de tipo de avión para un mismo número de vuelo. Realizar las siguientes consultas:

**1. Listar los números de vuelo de A hasta F.**

$\sigma \text{ nro-vuelo } (\pi \text{ desde=A, hasta=F}) \text{ (VUELOS)}$

$\Pi \text{ nro - vuelo } (\sigma \text{ nro - vuelo like `A \% F` }) \text{ (VUELO)}$

SELECT nro-vuelo FROM VUELOS WHERE desde ='A%' AND hasta='F%';

**2. Listar los tipos de avión que no son utilizados en ningún vuelo que pase por B.**

$\pi$  tipo-avión ( $\sigma$  destino = 'B') (AVION-UTILIZADO)(INFO-PASAJEROS)

$\Pi$  tipo – avión ( $\sigma$  diferente B )( avión – utilizado )

SELECT A.tipo-avión from AVION-UTILIZADO A, INFO-PASAJEROS I WHERE I.destino not in 'B';

**3. Listar los pasajeros y números de vuelo para aquellos pasajeros que viajan de A a D pasando por B.**

$\pi$  nombre, nro-vuelo ( $\sigma$  origen='A',  $\pi$  destino='D') (INFO-PASAJEROS)

$\Pi$  nombre , nro –vuelo ( $\sigma$ ( nombre like `A` % `D`) and (nombre like % `B`%))(INFO –PASAJERO))

SELECT P.nombre, V.nro-vuelo FROM INFO-PASAJEROS P, VUELOS V WHERE P.origen='A' AND P.destino='D'

**4. Listar los tipos de avión que son utilizados en todos los vuelos que pasan por C.**

$\pi$  tipo-avión ( $\sigma$  destino='C')(AVION-UTILIZADO)(INFO-PASAJEROS)

$\Pi$  tipo ( $\sigma$  tipo like % `c` %) (AVION –UTILIZADO)

SELECT A.tipo-avión FROM AVION-UTILIZADO 'A', INFO-PASAJEROS I WHERE I.destino='C';

**EJERCICIO 3**

Sea la siguiente Base de Datos:

**Empleado**(nro\_empleado,nombre,domicilio,localidad,antigüedad)  
**Funcion**(nro\_funcion,descripcion) Planta(nro\_planta,descripcion,localidad)  
**Tarea**(nro\_empleado, nro\_funcion) Organización(nro\_planta,nro\_funcion)  
**Trabaja**(nro\_empleado, nro\_planta)  
**Distancia**(localidad1, localidad2,distancia)

Escribir en Algebra Relacional la siguiente consulta:

Listar los nombres de los empleados que viven más lejos de su lugar de trabajo

$\pi$  nombre ( $\sigma$  distancia>todos) (EMPLEADO)(DISTANCIA)

SELECT NOMBRE FROM EMPLEADO WHERE DISTANCIA = > LOCALIDAD1;

#### **EJERCICIO 4**

Sea la siguiente Base de Datos:

**Artículos**(nro\_art,descripcion,peso,precio\_unit,fabricado\_en)  
**Facturas**(nro\_fact, nro\_cli, fecha, fecha\_venc)  
**DetalleFactura**(nro\_fact,nro\_art,cant)  
**Clientes**(nro\_cli, nombre, domicilio, ciudad, País, Ocupación)

**Expresar la siguiente consulta en Algebra Relacional:**

**Listar los nombres de los clientes que hayan comprado al menos 2 artículos distintos.**

$(\pi$  nombre ( $\sigma$  cant>2)\*  $\pi$  descripción)(CLIENTES, DETALLE FACTURA, ARTICULOS).

SELECT CLIENTES.NOMBRE FROM CLIENTES INNER JOIN FACTURAS ON  
FACTURAS.nro\_cli = CLIENTES.nro\_cli INNER JOIN DETALLES ON  
FACTURAS.nro\_fact = DETALLES.nro\_fact WHERE ARTICULOS.nro\_art > 2 AND  
ARTICULOS.nro\_art <> ARTICULOS.nro\_art

## EJERCICIO 5

Sea la siguiente Base de Datos:

**Personas**(nro\_doc, nombre, domicilio)  
**Supervisa\_a**(nro\_doc\_supervisor, nro\_doc\_supervisado)

Asumir que cada persona es supervisada a lo sumo por un supervisor.

**Escribir en AR la siguiente consulta: Listar los nombres de las personas que trabajan con Juan Perez (suponer que existe un único Juan Perez)**

$\pi$  nombre ( $\sigma$  nro-doc-supervisor='Juan Perez' (Supervisa)) (Personas)

SELECT P.nombre FROM Personas P, Supervisa S WHERE  
P.nro-doc=S.nro\_doc\_supervisado AND S.nro\_doc\_supervisor='Juan Perez';

## EJERCICIO 6

Sea la siguiente Base de Datos:

**Empleados**(nro\_e, nombre, domicilio, ciudad)  
**Asignado\_a**(nro\_e, cod\_tarea, cant\_horas)  
**Tareas**(cod\_tarea, desc\_tarea)

**Escribir en AR la siguiente consulta, sin utilizar operadores derivados: Listar los nombres de las personas asignadas a todas las tareas**

$\pi$  nombre ( $\sigma$  nro-e =(  $\sigma$  group by cod-tarea (Tareas) (Asignado))(Empleado))

SELECT E.nombre from Empleados E, Asignado A, Tareas T WHERE  
E.nro\_e=A.nro\_e AND A.cod\_tarea=T.cod tarea AND E.nro\_e in (SELECT  
T.cod\_tarea FROM Tareas GROUP BY cod\_tarea);

## EJERCICIO 7

Para la siguiente base de datos:

**ALUMNOS**(nroLeg,nombre,domicilio,planDeEstudios)

**MATERIAS**(nroMat,nombre,planDeEstudios)

**EXAMENES**(nroLeg,nroMat,nota,fecha)

**CURSAN**(nroLeg, nroMat)

**CORRELATIVA**(nroMat,nroCorrelat)

Expresar en Álgebra Relacional la siguiente consulta:

**Listar los nombres de los alumnos que no cursan ninguna materia de la cual hayan rendido su correlativa al menos dos veces .**

$\pi$  nombre ( $\sigma$  norCorrelat  $\geq 2$   
(CORRELATIVA)(MATERIAS)(EXAMENES)(ALUMNOS)(CURSAN)

SELECT A.nombre FROM ALUMNOS A, MATERIAS M, EXAMENES E, CURSAN C,  
CORRELATIVA CO WHERE A.nroLeg=E.nroLeg AND E.nroMat=M.nroMat AND  
M.nroMat=C.nroMat AND CO.norCorrelta  $\geq 2$ ;

## EJERCICIO 8

Sea la siguiente base de datos:

**ALUMNOS**(nroLeg,nombre,domicilio,planDeEstudios)

**MATERIAS**(nroMat,nombre,planDeEstudios)

**EXAMENES**(nroLeg,nroMat,nota,fecha)

**CURSAN**(nroLeg, nroMat)

Expresar en Álgebra Relacional la siguiente consulta:

**Listar los nombres de los alumnos que no cursan ninguna materia en la cual ningún alumno que rindió la materia al menos 2 veces obtuvo una nota mayor que 6 en un final.**



$\pi$  nombre ( $\sigma$  nroMat not in ( $\sigma$  nroLeg $\geq$ 2, nota $\geq$ 6  
(Tareas)(CURSAN)(MATERIAS)(EXAMENES)(ALUMNOS))

SELECT A.nombre FROM ALUMNOS A, MATERIAS M, EXAMENES E, CURSAN C  
WHERE A.nroLeg=E.nroLeg AND E.nroMat=M.nroMat AND M.nroMat=C.nroMat  
AND C.nroMat not in (SELECT nroMat FROM EXAMENES WHERE nroLeg $\geq$ 2  
AND nota $\geq$ 6);

## EJERCICIO 9

Para la siguiente base de datos:

**ALUMNOS**(dni, nombre,apellido,nacionalidad)  
**FACULTADES**(codfacultad, nombre)  
**CARRERAS**(codcarrera, nombre,codfacultad,cantidadDeAlumnos) codfacultad  
**foreign key references facultades**(codfacultad)  
**ESTUDIANTEDE**(dni,codcarrera)

Expresar en Algebra Relacional la consulta:

**Listar los nombres de los estudiantes que no estudian carreras que se dictan en facultades en las que estudia algún estudiante español.**

$\pi$  nombre ( $\sigma$  codfacultad not in ( $\sigma$  nacionalidad="Español")  
(FACULTAD)(CARRERAS)(EXAMENES)(ALUMNOS))

SELECT A.nombre FROM ALUMNOS A, CARRERAS C, FACULTADES F,  
ESTUDIANTES E WHERE A.dni=E.dni AND E.codcarrera=C.codcarrera AND  
C.codfacultad=F.codfacultad AND F.codfacultad not in (SELECT F.codfacultad  
FROM ALUMNOS A, CARRERAS C, FACULTADES F, ESTUDIANTES E WHERE  
A.dni=E.dni AND E.codcarrera=C.codcarrera AND C.codfacultad=F.codfacultad  
AND A.nacionalidad="Español")

## Ejercicio 10

sea la siguiente base de datos:

**ALUMNO** (nro\_leg,nombre,domicilio,plan\_de\_estudios)  
**CURSOS**(nro\_curso,nombre,horario)  
**MATERIA**(nro\_mat,nombre,plan\_de\_estudio)  
**APROBO**(nro\_leg,nro\_mat,nota)

Expresa las siguientes consultas en algebra Relacional:

- 1. Lista los nombre de los alumnos que aprobaron solamente materias correspondiente a su plan de estudios.**

$\pi$  nombre ( $\sigma$  nroleg in ( $\pi$  nroleg (APROBO))(MATERIAS)(ALUMNOS))

SELECT A.nombre FROM ALUMNOS A, MATERIAS M, APROBO AP WHERE  
A.plan\_de\_estudios=M.plan\_de\_estudios AND M.nro\_mat=AP.nro\_mat AND  
A.nroleg in (SELECT nroleg FROM APROBO);

- 2. lista las materias aprobadas con un 9 por al menos 2 alumnos que no hayan cursado ninguna materia del plan 96.**

$\pi$  nombre ( $\sigma$  nota=9(APROBO))( $\sigma$  plan\_de\_estudios=96(ALUMNOS))(MATERIAS))

SELECT M.nombre FROM MATERIAS M, ALUMNOS A, APROBO AP WHERE  
M.plan\_de\_estudios=A.plan\_de\_estudios AND A.nroleg=AP.nroleg AND AP.nota=9  
AND A.plan\_de\_estudios=96;

- 3. listar las materias no aprobadas por ningun alumno que haya obtenido mas de 8 puntos en alguna materia correspondiente a su mismo plan de estudios.**

$\pi$  nombre ( $\sigma$  nroleg in ( $\pi$  nroleg ( $\sigma$  nota>8)(MATERIAS)(ALUMNOS)(APROBO)))

SELECT M.nombre FROM MATERIAS M, ALUMNOS A, APROBO AP, WHERE  
M.nro\_mat=AP.nro\_mat AND AP.plan\_de\_estudios=A.plan\_de\_estudios AND  
A.nroleg in (SELECT nroleg FROM APROBO WHERE nota>8);

## Ejercicio 11

Dado la base de datos:

**FRECUENTA**(nombre-pers,nombre-bar)

**SIRVE**(nombre-bar, nombre-cerveza)

**GUSTA**(nombre-persona,nombre-cerveza)

- 1. Encontrar las personas que frecuentan un bar que sirve una cerveza que les gusta.**

$\pi$  nombre FRECUENTA ( $\sigma$  G.NOMBRE-CERVEZA = S.CERVEZA)

SELECT DISTINCT FAUX.NOMBRE-PERS FROM FRECUENTA AS F,  
SIRVE AS S, GUSTA AS G, FRECUENTA AS FAUX WHERE G.NOMBRE  
-CERVEZA = S.CERVEZA;

- 2. Encontrar las personas que beben en el mismo bar que las personas a las que les gusta la cerveza "Quilmes".**

$\pi$  nombre FRECUENTA ( $\sigma$  G.NOMBRE-CERVEZA = 'Quilmes')  $\bowtie$   
FRECUENTA  $\bowtie$  GUSTA FRECUENTA.NOMBRE-PERS =  
GUSTA.NOMBRES-PERS  $\bowtie$  SIRVE SIRVE.NOMBRE-CERVEZA =  
GUSTA.NOMBRE-CERVEZA

SELECT DISTINCT FAUX.NOMBRE-PERS FROM FRECUENTA AS F,  
SIRVE AS S, GUSTA AS G, FRECUENTA AS FAUX WHERE G.NOMBRE  
-CERVEZA = 'Quilmes' AND S.NOMBRE-CERVEZA =  
G.NOMBRE-CERVEZA AND F.NOMBRE-BAR = S.NOMBRE-BAR AND  
F.NOMBRE-PERS = G.NOMBRES-PERS AND FAUX.NOMBRE-BAR =  
F.NOMBRE-BAR;

- 3. Encontrar las personas que beben en el mismo bar que aquellas a las que le gusta una marca de cerveza que sirva dicho bar y que le guste a Juan Perez.**

$\pi$  nombre FRECUENTA ( $\sigma$  F.NOMBRE-PERS = 'Juan Perez')  $\bowtie$  FRECUENTA  
 $\bowtie$  GUSTA FRECUENTA.NOMBRE-PERS = GUSTA.NOMBRES-PERS  $\bowtie$   
SIRVE SIRVE.NOMBRE-CERVEZA = GUSTA.NOMBRE-CERVEZA  
( $\sigma$ G.NOMBRE-CERVEZA = S..NOMBRE-CERVEZA ) ( $\sigma$ F.NOMBRE-BAR =  
S.NOMBRE-BAR )

SELECT DISTINCT FAUX.NOMBRE-PERS FROM FRECUENTA AS F,  
SIRVE AS S, GUSTA AS G, FRECUENTA AS FAUX WHERE G.NOMBRE  
-CERVEZA IN ( SELECT NOMBRE-CERVEZA FROM GUSTA WHERE  
NOMBRE-PERS = 'Juan Perez') AND S.NOMBRE-CERVEZA =  
G.NOMBRE-CERVEZA AND F.NOMBRE-BAR = S.NOMBRE-BAR AND  
F.NOMBRE-PERS = G.NOMBRE-PERS AND FAUX.NOMBRE-PERS <>  
F.NOMBRE-PERS AND FAUX.NOMBRE-BAR = F.NOMBRE-BAR;

**Encontrar las personas que frecuentan solamente bares que sirven alguna cerveza que les guste.(asumir que cada persona frecuenta al menos un bar y le gusta al menos una cerveza)**

$\pi$  nombre GUSTA ( $\sigma$  F.NOMBRE-BAR = G.NOMBRE-BAR)  $\bowtie$  FRECUENTA  $\bowtie$   
 GUSTA FRECUENTA.NOMBRE-PERS = GUSTA.NOMBRES-PERS  $\bowtie$  SIRVE  
 SIRVE.NOMBRE-CERVEZA = GUSTA.NOMBRE-CERVEZA  
 ( $\sigma$ G.NOMBRE-CERVEZA = S..NOMBRE-CERVEZA ) ( $\sigma$ F.NOMBRE-BAR =  
 S.NOMBRE-BAR )

SELECT DISTINCT G.NOMBRE-PERS FROM FRECUENTA AS F, SIRVE AS  
 S, GUSTA G WHERE NOMBRE-PERS = G.NOMBRES-PERS AND  
 F.NOMBRE-BAR = S.NOMBRE-BAR AND G.NOMBRE-CERVEZA =  
 S.NOMBRE-CERVEZA GROUP BY G.NOMBRE-PERS HAVING COUNT  
 (F.NOMBRE-PERS) = (SELECT COUNT (F1.NOMBRE-BAR) FROM  
 FRECUENTA F1 WHERE G.NOMBRE-PERS = F1.NOMBRE-PERS);

**4. Encontrar las personas que no frecuentan ningún bar que sirva alguna cerveza que le guste.**

$\pi$  nombre FRECUENTA  $\bowtie$  FRECUENTA  $\bowtie$  GUSTA  
 FRECUENTA.NOMBRE-PERS = GUSTA.NOMBRES-PERS  $\bowtie$  SIRVE  
 SIRVE.NOMBRE-CERVEZA = GUSTA.NOMBRE-CERVEZA  
 ( $\sigma$ G.NOMBRE-CERVEZA  $\neq$  S..NOMBRE-CERVEZA ) ( $\sigma$ F.NOMBRE-BAR  
 $\neq$  S.NOMBRE-BAR )

SELECT DISTINCT F1.NOMBRE-PERS FROM FRECUENTA AS F WHERE  
 NOT EXIST (SELECT FAUX.NOMBRE-PERS FROM FRECUENTA AS FAUX,  
 SIRVE AS S, GUSTA AS G WHERE S.NOMBRE-CERVEZA =  
 G.NOMBRE-CERVEZA AND FAUX.NOMBRE-BAR = F.NOMBRE-BAR AND  
 FAUX.NOMBRE-PERS = G.NOMBRE-PERS);

**EJERCICIO 12**

Sea la BD:

**PERSONAS**(tipo-doc,num-doc,nomyap,dir,tel,fnac,sexo)

**PROGENITOR**(tipo-doc,num-doc,tipo-doc-hijo,num-doc-hijo)

**1. Listar para cada Juan Perez los tipo y número de documento, nombre y apellido y teléfonos de todos sus hijos.**

$\pi$  nomyap, tipo-doc ( $\sigma$  nomyap="Juan Perez"(PERSONAS)(PROGENITOR))

SELECT P.nomyap, PR.tipo-doc, PR.num-doc FROM PERSONAS P, PROGENITOR  
 PR WHERE P.tipo-doc=PR.tipo-doc AND P.nomyap="Juan Perez" GROUP BY  
 P.nomyap.

**2. b.-Idem a ,de :**

**1. todos sus hermanos ( los hijos de su padre y/o su madre).**

$\pi$  nomyap, tipo-doc ( $\sigma$  nomyap="Juan Perez"( $\sigma$  sexo="h" U  $\sigma$  sexo="m"  
(PERSONAS)(PROGENITOR)))

SELECT P.nomyap, PR.tipo-doc, PR.num-doc FROM PERSONAS P,  
PROGENITOR PR WHERE P.tipo-doc=PR.tipo-doc AND  
P.nomyap="Juan Perez" AND (P.sexo="h" OR P.sexo="m")GROUP BY  
P.nomyap.

**2. su madre.**

$\pi$  nomyap, tipo-doc ( $\sigma$  nomyap="Juan Perez"( PERSONAS)( $\sigma$   
tipo-doc="madre"PROGENITOR)))

SELECT P.nomyap, PR.tipo-doc, PR.num-doc FROM PERSONAS P,  
PROGENITOR PR WHERE P.tipo-doc=PR.tipo-doc AND  
P.nomyap="Juan Perez" AND PR.tipo-doc="madre"GROUP BY  
P.nomyap.

**3. su abuelo materno.**

$\pi$  nomyap, tipo-doc ( $\sigma$  nomyap="Juan Perez"( PERSONAS)( $\sigma$   
tipo-doc="abuelo materno"PROGENITOR)))

SELECT P.nomyap, PR.tipo-doc, PR.num-doc FROM PERSONAS P,  
PROGENITOR PR WHERE P.tipo-doc=PR.tipo-doc AND  
P.nomyap="Juan Perez" AND PR.tipo-doc="abuelo materno"GROUP  
BY P.nomyap.

**4. todos sus nietos.**

$\pi$  nomyap, tipo-doc ( $\sigma$  nomyap="Juan Perez"( PERSONAS)( $\sigma$   
tipo-doc="abuelo materno"PROGENITOR)))

SELECT P.nomyap, PR.tipo-doc, PR.num-doc FROM PERSONAS P,  
PROGENITOR PR WHERE P.tipo-doc=PR.tipo-doc AND  
P.nomyap="Juan Perez" AND PR.tipo-doc="nieto"GROUP BY  
P.nomyap.

**EJERCICIO 13**

Una oficina gubernamental desea construir un complejo habitacional, para lo cual elaboró la siguiente Base de Datos:

**TRAMO**( c\_ciudadA, c\_ciudadB, Distancia).

**CIUDADES**( c\_ciudad, nombre, cant\_escuelas, cant\_fábricas).

Para decidir dónde instalarlo, desea conocer los siguientes datos:

- 1. Las ciudades alcanzables desde la ciudad con mayor cantidad de fábricas, recorriendo no más de 2 tramos, c/u de los cuales no puede tener más de 10Km de longitud.**

$\pi$  nombre ( $\sigma$  ciudadB $\leq$ 2, distancia $\leq$ 10(TRAMO))( $\sigma$  max(cant\_fabricas)(CIUDADES)))

SELECT C.nombre FROM TRAMO T, CIUDADES C WHERE  
C.cant\_fabricas=(SELECT MAX(cant\_fabricas) FROM CIUDADES)  
T.ciudadB $\leq$ 2 AND T.distancia $\leq$ 10;

- 2. Las ciudades con más de 10 fábricas, que estén conectadas en forma directa con todas las demás, siempre que ningún tramo supere los 50 Km.**

$\pi$  nombre ( $\sigma$  ciudadB $\leq$ 2, distancia $\leq$ 10(TRAMO))(CIUDADES))

SELECT C.nombre FROM TRAMO T, CIUDADES C WHERE C.cant\_fabricas  
>10 AND T.ciudadB=21 AND T.distancia $\leq$ 50;

- 3. Los pares de ciudades, de la forma (ciudad1,ciudad2), que son alcanzables a través de, a lo sumo, otra ciudad, llamemos a ésta, ciudad3 (esto es, encontrar lo caminos ciudad1-ciudad3-ciudad2).**

$\pi$  nombre ( $\sigma$  ciudadA=1, ciudadB=1(TRAMO))(CIUDADES))

SELECT T.ciudadA, T.ciudadB, C.nombre FROM TRAMO T, CIUDADES C  
WHERE T.ciudadA=1 AND T.ciudadB=1 ;

## **EJERCICIO 14**

Sea la siguiente Base de Datos:

**Artículos**(nro\_art,descripcion,color,peso,precio\_unit,fabricado\_en)

**Factura**(nro\_fact,nro\_cli,fecha, fecha\_venc)

**DetalleFactura**(nro\_fact,nro\_art,cant)

**Clientes**(nro\_cli, nombre, domicilio, ciudad, País, Ocupación)

Escribir en AR y SQL la siguiente consulta:

**Listar la descripción de los artículos que no fueron facturados a ningún cliente al que se le facturó más de 2(DOS) veces algún artículo de color rojo.**

$\pi$  descripcion ( $\sigma$  nro\_art not in( $\pi$  nro\_cli  
(FACTURA)(CLIENTE)(DETALLE)(ARTICULOS))

SELECT A.descripcion FROM Artículos A, Factura F, Detalle D, Clientes C WHERE  
A.nro\_art=D.nro\_art AND D.nro\_fact=F.nro\_fact AND F.nro\_cli=C.nro\_cli AND  
nro\_art NOT IN (SELECT nro\_cli FROM Factura);