

Soluciones Ejercicio modelo relacional: Álgebra y Cálculo Relacional

Ejercicio 1

Suponer el siguiente modelo relacional de una empresa con los siguientes esquemas de relación:

Empleado(nombre, inicial, apellido, nss, fecha_nacimiento, dirección, sexo, salario, nss_supervisor, nd)

Departamento(nombre, numerod, nss_jefe, fecha_inic_jefe)

Localizaciones_dept(numerod, localizaciond)

Trabaja_en(nsse, np, horas)

Proyecto(nombrep, numerop, localizacionp, nd)

Dependiente(nsse, nombre_dependiente, sexo, fecha_ncto, parentesco)

Consultas a resolver en el Álgebra Relacional:

1. Obtener el nombre y la dirección de todos los empleados que trabajan para el departamento 'Investigación'.

Temp1 $\leftarrow \sigma_{\text{nombredp}='Investigacion'}(\text{DEPARTAMENTO})$

Temp2 $\leftarrow \text{Temp1} \bowtie_{\text{Nd=numerod}} \text{EMPLEADO}$

$\Pi_{\text{nombre, apellido, direccion}}(\text{Temp2})$

2. Para cada proyecto localizado en 'Santiago', obtener una lista con el número de proyecto, el número de departamento que lo controla, y el apellido, la dirección y la fecha de nacimiento del jefe de dicho departamento.

Temp1 $\leftarrow \sigma_{\text{localizacionp}='Santiago'}(\text{PROYECTO})$

Temp2 $\leftarrow \text{Temp1} \bowtie_{\text{Nd=numerod}} \text{DEPARTAMENTO}$

Temp3 $\leftarrow \text{Temp2} \bowtie_{\text{NSS=NSS_JEFE}} \text{EMPLEADO}$

$\Pi_{\text{nombre, apellido, direccion}}(\text{Temp3})$

3. Buscar el nombre de los empleados que trabajan en todos los proyectos controlados por el departamento número 5.

Temp1 $\leftarrow \Pi_{\text{Numerop}}(\sigma_{\text{numd}=5}(\text{PROYECTO}))$

Temp2 $\leftarrow \Pi_{\text{NSSE, NP}}(\text{TRABAJA_EN})$

Temp3 $\leftarrow \text{Temp2} \div \text{Temp1}$

$$\Pi_{\text{nombre, apellido}} (\text{EMPLEADO} \bowtie_{\text{NSS=NSS_JEFE}} \text{Temp3})$$

4. Obtener una lista con los números de los proyectos en que intervienen un empleado cuyo apellido es 'Smith', ya sea como trabajador o como jefe del departamento que controla el proyecto.

$$\text{Temp1} \leftarrow \Pi_{\text{NSS}} (\sigma_{\text{Apellido}='Smith'}(\text{EMPLEADO}))$$

$$\text{Temp2} \leftarrow \Pi_{\text{NP}} (\text{TRABAJA_EN} \bowtie_{\text{NSSE=NSS}} \text{Temp1})$$

$$\text{Temp3} \leftarrow \Pi_{\text{Apellidos, numerod}} (\text{EMPLEADO} \bowtie_{\text{NSS=NSS_JEFE}} \text{DEPARTAMENTO})$$

$$\text{Temp4} \leftarrow \Pi_{\text{numerod}} (\sigma_{\text{Apellido}='Smith'}(\text{Temp3}))$$

$$\text{Temp5} \leftarrow \Pi_{\text{Nuerop}} (\text{PROYECTOS} \bowtie_{\text{ND=numerod}} \text{Temp4})$$

$$\text{Temp2} \cup \text{Temp5}$$

5. Obtener una lista con los nombres de todos los empleados que tienen dos ó más personas dependientes de ellos.

$$\text{Temp1} \leftarrow \Pi_{\text{NSS}} \text{G}_{\text{count(nombre_dependiente) as cuenta}} (\text{DEPENDIENTE})$$

$$\text{Temp2} \leftarrow \sigma_{\text{cuenta} \geq 2} (\text{Temp1})$$

$$\Pi_{\text{nombre, apellido}} (\text{Temp2} \bowtie_{\text{NSSE=NSS}} \text{EMPLEADO})$$

6. Obtener el nombre de los empleados que no tienen otras personas dependientes de ellos.

$$\text{Temp1} \leftarrow \rho_{\text{dep(NSS)}} (\Pi_{\text{NSSE}}(\text{DEPENDIENTES}))$$

$$\text{Temp2} \leftarrow \Pi_{\text{NSS}} (\text{EMPLEADO})$$

$$\text{Temp3} \leftarrow \text{Temp2} - \text{Temp1}$$

$$\Pi_{\text{nombre, apellido}} (\text{Temp3} \bowtie \text{EMPLEADO})$$

7. Obtener el nombre de los jefes que tienen por lo menos una persona dependiente de ellos.

$$\text{Temp1} \leftarrow \rho_{\text{dep(NSSE)}} (\Pi_{\text{NSS_JEFE}}(\text{DEPARTAMENTO}))$$

$$\text{Temp2} \leftarrow \Pi_{\text{NSSE}} (\text{DEPENDIENTE})$$

$$\text{Temp3} \leftarrow \text{Temp2} \cap \text{Temp1}$$

$\Pi_{\text{nombre, apellido}} (\text{Temp3} \bowtie_{\text{NSS=NSSE}} \text{EMPLEADO})$

Consultas a resolver en el Cálculo Relacional de Tuplas:

1. Obtener la fecha de nacimiento y la dirección del empleado cuyo nombre sea 'John B. Smith'

$\{ t \mid \exists s \in \text{empleado} (t[\text{fecha_ncto}] = s[\text{fecha_ncto}] \wedge t[\text{direccion}] = s[\text{direccion}] \wedge s[\text{nombre}] = \text{'John'} \wedge s[\text{inic}] = \text{'B'} \wedge s[\text{apellido}] = \text{'Smith'}) \}$

2. Obtener el nombre y la dirección de todos los empleados que trabajan para el departamento 'Investigación'.

$\{ t \mid \exists s \in \text{empleado} (t[\text{nombre}] = s[\text{nombre}] \wedge t[\text{apellido}] = s[\text{apellido}] \wedge \exists d \in \text{departamento} (d[\text{numerod}] = s[\text{nd}] \wedge d[\text{nombred}] = \text{'Investigacion'})) \}$

3. Buscar el nombre de los empleados que trabajan en algún proyecto controlado por el departamento número 5.

$\{ t \mid \exists e \in \text{empleado} (t[\text{apellido}] = e[\text{apellido}] \wedge t[\text{nombre}] = e[\text{nombre}] \wedge (\exists x \in \text{proyecto} (x[\text{numd}] = 5 \wedge \exists w \in \text{trabaja_en} (w[\text{nsse}] = e[\text{nsse}] \wedge x[\text{numerop}] = w[\text{np}])))) \}$

4. Obtener una lista con los números de los proyectos en que intervienen un empleado cuyo apellido es 'Smith', ya sea como trabajador o como jefe del departamento que controla el proyecto.

$\{ t \mid \exists p \in \text{proyecto} (t[\text{numerop}] = p[\text{numerop}] \wedge (\exists e \in \text{empleado} (e[\text{apellido}] = \text{'Smith'} \wedge \exists w \in \text{trabaja_en} (w[\text{np}] = p[\text{numerop}] \wedge w[\text{nsse}] = e[\text{nsse}])) \vee \exists m \in \text{empleado} (m[\text{apellido}] = \text{'Smith'} \wedge \exists d \in \text{departamento} (p[\text{numd}] = d[\text{numerod}] \wedge d[\text{nss_jefe}] = m[\text{nss}])))) \}$

5. Buscar el nombre de los empleados que trabajan en todos los proyectos controlados por el departamento número 5.

$\{ t \mid \exists e \in \text{empleado} (t[\text{apellido}] = e[\text{apellido}] \wedge t[\text{nombre}] = e[\text{nombre}] \wedge \forall x \in \text{proyecto} (x[\text{numd}] = 5 \Rightarrow \exists w \in \text{trabaja_en} (w[\text{nsse}] = e[\text{nsse}] \wedge x[\text{numerop}] = w[\text{np}]))) \}$

6. Obtener el nombre de los empleados que no tienen otras personas dependientes de ellos.

$\{ t \mid \exists e \in \text{empleado} (t[\text{apellido}] = e[\text{apellido}] \wedge t[\text{nombre}] = e[\text{nombre}] \wedge \neg \exists d \in \text{dependiente} (e[\text{nss}] = d[\text{nsse}])) \}$

7. Obtener el nombre de los jefes que tienen por lo menos una persona dependiente de ellos.

$\{ t \mid \exists e \in \text{empleado} (t[\text{nombre}] = e[\text{nombre}] \wedge t[\text{apellido}] = e[\text{apellido}] \wedge (\exists d \in \text{departamento} (\exists p \in \text{dependiente} (e[\text{nss}] = p[\text{nss_jefe}] \wedge p[\text{nsse}] = e[\text{nss}])))) \}$

Consultas a resolver en el Cálculo Relacional de Dominios.

1. Obtener la fecha de nacimiento y la dirección del empleado cuyo nombre sea 'John B. Smith'

$$\{ \langle u, v \rangle \mid \exists q, r, s (\langle q, r, s, t, u, v, w, x, y, z \rangle \in \text{empleado} \wedge q = \text{'John'} \wedge r = \text{'B'} \wedge s = \text{'Smith'}) \}$$

2. Obtener el nombre y la dirección de todos los empleados que trabajan para el departamento 'Investigación'.

$$\{ \langle q, s, v \rangle \mid \exists z (\langle q, r, s, t, u, v, w, x, y, z \rangle \in \text{empleado} \wedge \exists l, m (\langle l, m, n, o \rangle \in \text{departamento} \wedge l = \text{'Investigacion'} \wedge m = z)) \}$$

3. Obtener el nombre de los empleados que no tienen otras personas dependientes de ellos.

$$\{ \langle q, s \rangle \mid \exists t (\langle q, r, s, t, u, v, w, x, y, z \rangle \in \text{empleado} \wedge (\neg \exists l (\langle l, m, n, o, p \rangle \in \text{dependiente} \wedge t = l))) \}$$

4. Obtener el nombre de los jefes que tienen por lo menos una persona dependiente de ellos.

$$\{ \langle s, q \rangle \mid \exists t (\langle q, r, s, t, u, v, w, x, y, z \rangle \in \text{empleado} \wedge (\exists j (\langle h, i, j, k \rangle \in \text{departamento} \wedge \exists l (\langle l, m, n, o, p \rangle \in \text{dependiente} \wedge t = j \wedge l = t)))) \}$$

Cuestión 4**[15 p]**

Una empresa dispone de un sistema para manejar los proyectos que realiza y el personal asignado a os mismos. Se usan las siguientes tablas:

PROYECTOS(cod_proy, cod_cliente, fecha_inicio, tipo)

PERSONAL(cod_func, nombre, fecha_ingreso)

TAREAS(cod_tarea, descripción, tipo)

ASIGNACION(cod_func, cod_proy, cod_tarea)

REGISTRO_HORAS(cod_func, cod_proy, fecha, cant_horas)

Se pide:

a) Resolver las siguientes consultas utilizando el álgebra relacional

- 1) Obtener los códigos de los proyectos, clientes y tipo de los proyectos que sólo tienen asignadas personas que ingresaron en el 2005 en la empresa.

Temp1 $\leftarrow \sigma_{\text{fecha_ingreso} \geq '01-01-1995' \wedge \text{fecha_ingreso} \leq '31-12-1995'}(\text{PERSONAL})$

Temp2 $\leftarrow \text{ASIGNACION} \bowtie \text{Temp1}$

Temp3 $\leftarrow \text{PROYECTOS} \bowtie \text{Temp2}$

Temp4 $\leftarrow \Pi_{\text{cod_proy}, \text{cod_cliente}, \text{tipo}}(\text{Temp3})$

- 2) Obtener los nombres de los funcionarios que se encuentran asignados a todos los proyectos de tipo consultoría o que tienen asignadas todas las tareas de Gestión.

Sola1 $\leftarrow \Pi_{\text{cod_proy}}(\sigma_{\text{tipo}='consultoria'}(\text{PROYECTOS}))$

Sola2 $\leftarrow \Pi_{\text{cod_func}, \text{cod_proy}}(\text{ASIGNACION}) \div \text{Sola1}$

Solb1 $\leftarrow \Pi_{\text{cod_tarea}}(\sigma_{\text{tipo}='gestion'}(\text{TAREA}))$

Solb2 $\leftarrow \Pi_{\text{cod_func}, \text{cod_tarea}}(\text{ASIGNACION}) \div \text{Solb1}$

Sol1 $\leftarrow \text{Sola1} \cup \text{Sola2}$

Sol $\leftarrow \Pi_{\text{nombre}}(\text{Sol1} \bowtie \text{PERSONA})$

- b) Resolver en el cálculo relacional la consulta: “Obtener el código de los funcionarios que no están asignados a proyectos del tipo Desarrollo, pero sí a tareas de tipo Desarrollo”.

$\{ t \mid \exists s \in \text{personal} (t[\text{cod_func}] = s[\text{cod_func}] \wedge \exists u \in \text{tareas} (u[\text{tipo}] = \text{'Desarrollo'} \wedge \exists a \in \text{asignacion} (a[\text{cod_func}] = s[\text{cod_func}] \wedge a[\text{cod_tarea}] = u[\text{cod_tarea}])) \wedge \neg (\exists p \in \text{proyectos} (p[\text{tipo}] = \text{'Desarrollo'} \wedge \exists a2 \in \text{asignacion} (a2[\text{cod_func}] = s[\text{cod_func}] \wedge a2[\text{cod_proy}] = p[\text{cod_proy}])))) \}$

Cuestión 3**[10 p]**

Dada la siguiente consulta expresada en lenguaje SQL:

```
SELECT DISTINCT C.nro_cli, C.nombre, E.fecha_inicio FROM Clientes C, Estancias E, Casas A
WHERE C.nro_cli=E.nro_cliente AND E.nro_casa = A.nro_casa and A.categoria = 'A' AND E.dias>5
```

Se pide:

- a) Obtener una expresión equivalente del álgebra relacional que utilice sólo operadores básicos o fundamentales del álgebra relacional.

$$\pi_{C.nro_cli, C.nombre, E.fecha_inicio}(\sigma_{E.nro_casa=A.nro_casa \wedge C.nro_cli=E.nro_cliente \wedge A.categoria=A \wedge E.dias>5}(\rho_C(Clientes) \times \rho_E(Estancias) \times \rho_A(Casas)))$$

- b) Reescriba la expresión anterior utilizando obligatoriamente operadores adicionales que proporcionen el mismo resultado que la expresión anterior.

$$\pi_{C.nro_cli, C.nombre, E.fecha_inicio}(\sigma_{A.categoria=A \wedge E.dias>5}((\rho_C(Clientes) \bowtie_{C.nro_cli=E.nro_cliente} \rho_E(Estancias)) \bowtie \rho_A(Casas)))$$

Problema 1

[20 p]

Se tiene una base de datos con las siguientes relaciones:

RESIDENCIAS(cod-res, nom-res, cod-ins, año-constr, arquitecto, num-hab)

Contiene los datos de todas las residencias y cuyos atributos son: código de la residencia, nombre de la residencia, código de la institución que la construyó, año de construcción, arquitecto que la proyectó y número de habitaciones.

INSTITUCIONES(cod-ins, nom-ins, cant-becas)

Contiene los datos de las instituciones que construyeron residencias ya sean departamentales o particulares y cuyos atributos son: código de la institución, nombre de la institución y cantidad de becas que otorga.

DEPARTAMENTALES(cod-ins, departamento, programa, encargado, tel-contacto)

Contiene los proyectos de residencias construidas por organismos departamentales y cuyos atributos son: código de la institución. Departamento, programa o proyecto, encargado del proyecto y teléfono de contacto. Las instituciones que se encuentran en esta tabla, también están en INSTITUCIONES.

PARTICULARES(cod-ins, nom-fundador, encargado, tel-contacto)

Contiene los proyectos de residencias de fundaciones particulares y cuyos atributos son: código de la institución. Departamento, programa o proyecto, encargado del proyecto y teléfono de contacto. Las instituciones que se encuentran en esta tabla, también están en INSTITUCIONES pero no en DEPARTAMENTALES.

ESTUDIANTES(DNI, nombre, depto-origen, edad)

Contiene los datos de todos los estudiantes que se han alojado en alguna residencia y cuyos atributos son: DNI del estudiante, nombre del estudiante, departamento del que viene y edad.

EST_FAC(DNI, facultad, año-curso, promedio)

Contiene en qué facultades estudia cada estudiante. Sus atributos son: DNI del estudiante, nombre de la facultad, año que está cursando y promedio de notas. Cada estudiante de esta relación

CONTRATOS(cod-res, DNI, año, becado, habitación, teléfono)

Contiene en qué residencia se ha alojado cada estudiante en cada año, ya que los contratos de alojamiento son anuales. Sus atributos son: código de residencia, DNI del estudiante, año del contrato, indicación de existencia de beca (SI ó NO), habitación que ocupa y teléfono de la habitación. Las residencias y estudiantes de esta tabla se encuentran en RESIDENCIAS y ESTUDIANTES respectivamente.

Se pide:

- a) Escribir en Algebra Relacional la consulta: “Listar el código de institución y teléfono de contacto de todas las instituciones que otorgan 10 o más becas”.

Temp1 $\leftarrow \pi_{\text{cod-ins, tel-contacto}}(\text{DEPARTAMENTALES}) \cup \pi_{\text{cod-ins, tel-contacto}}(\text{PARTICULARES})$

Temp2 $\leftarrow \pi_{\text{cod-ins}}(\sigma_{\text{cant_becas} \geq 10}(\text{INSTITUCIONES}))$

Resultado $\leftarrow A \bowtie B$

- b) Escribir en Algebra Relacional la consulta: “Devolver los códigos de las residencias construidas por organismos departamentales que sólo han alojado a estudiantes del mismo departamento que el organismo departamental”.

Temp1 $\leftarrow \pi_{\text{cod-res, departamento}}(\text{RESIDENCIAS} \bowtie \text{DEPARTAMENTALES})$

Temp2 $\leftarrow \sigma_{\text{departamento} \neq \text{depto_origen}}(\text{Temp1} \bowtie \text{CONTRATOS} \bowtie \text{ESTUDIANTES})$

Resultado $\leftarrow \pi_{\text{cod-res}}(\text{TEMP1}) - \pi_{\text{cod-res}}(\text{TEMP2})$

Solución Ejercicios modelo relacional SQL

Ejercicio 1

Suponer el siguiente modelo relacional de una empresa con os siguientes esquemas de relación:

Empleado(nombre, inicial, apellido, nss, fecha_nacimiento, dirección, sexo, salario, nss_supervisor, nd)

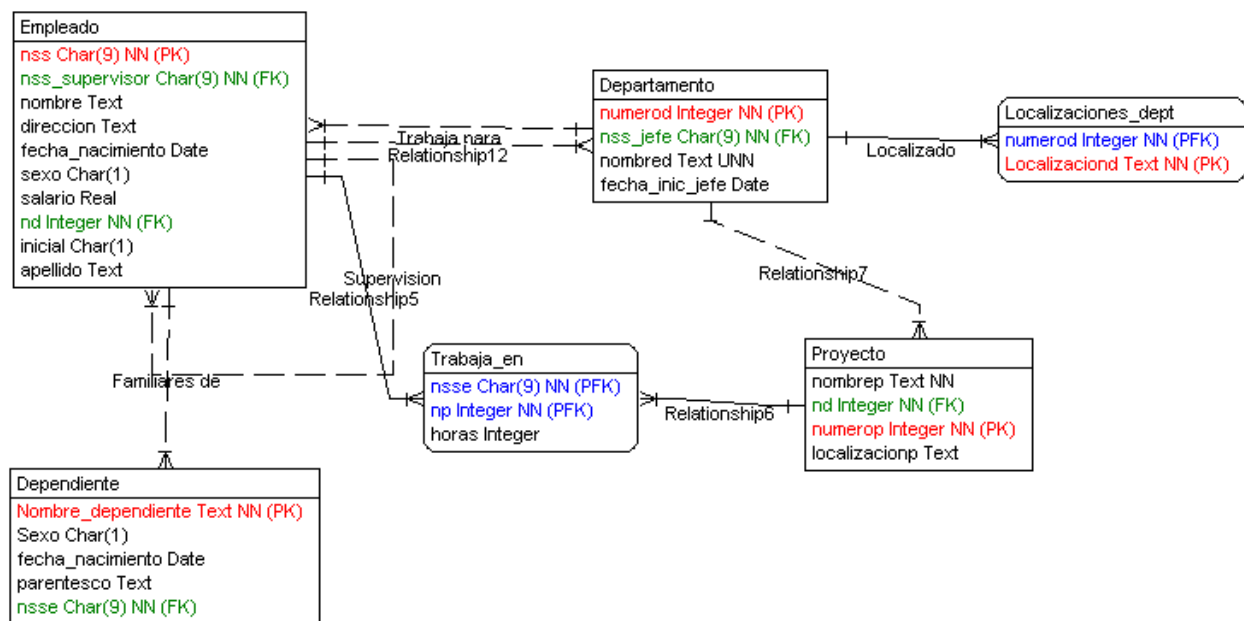
Departamento(nombred, numerod, nss_jefe fecha_inic_jefe)

Localizaciones_dept(numerod, localizaciond)

Trabaja_en(nsse, np, horas)

Proyecto(nombrep, numerop, localizacionp, nd)

Dependiente(nsse, nombre_dependiente, sexo, fecha_ncto, parentesco)



Consultas a resolver en el lenguaje SQL:

1. Obtener el nombre y la dirección de todos los empleados que trabajan para el departamento 'Investigación'.

```
Select nombre,direccion
from empleado,departamento
where nd=numerod and nombred='investigacion';
```

Otra forma más eficiente:

```
Select nombre,dirección
From empleado inner join departamento on nd=numerod
Where nombred='Investigacion';
```

2. Para cada proyecto localizado en 'Santiago', obtener una lista con el número de proyecto, el número de departamento que lo controla, y el apellido, la dirección y la fecha de nacimiento del jefe de dicho departamento.

```
Select numerop,nd,apellido,direccion,fecha_nacimiento
From proyecto,departamento,empleado
Where nd=numerod and nss_jefe=nss and localizacionp='Santiago';
Otra forma más eficiente:
```

```
Select a.numerop, a.nd, apellido, direccion, fecha_nacimiento
From
  ( Select numerop,nd,nss_jefe
    From proyecto inner join departamento
    On nd=numerod
    Where localizacionp='Santiago') as a
  Inner join empleado
  On nss_jefe=nss;
```

3. Buscar el nombre de los empleados que trabajan en todos los proyectos controlados por el departamento número 5.

Una división:

```
Select nombre,apellido
From empleado,(Select nsse
  From trabaja_en
  EXCEPT
  Select nsse from
  (Select nsse,numerop
  From (Select nsse from trabaja_en) as a,
      (Select numerop from proyecto where numero_dep=5) as b
  EXCEPT
  (Select nss,numerop from trabaja_en)) as c) as d
Where d.nss=empleado.nss;
```

Otra forma:

```
Select nombre,apellido from
  (Select nsse from trabaja_en natural join proyecto
   where numero_dep=5
  group by nsse
  having count(*) = (Select count(*) from proyecto where numerodep=5)) as t
  inner join empleado
  on nss=nsse;
```

4. Obtener una lista con los números de los proyectos en que intervienen un empleado cuyo apellido es 'Smith', ya sea como trabajador o como jefe del departamento que controla el proyecto.

```
(Select np from trabaja_en,Empleado
  Where nss=nsse and apellido='Smith')
UNION
(Select numerop as np from proyecto,departamento,empleado
  Where nd=numerod and nss_jefe=nss and apellido='Smith');
```

5. Obtener una lista con los nombres de todos los empleados que tienen dos ó más personas dependientes de ellos.

```
Select nombre,apellido
From empleado,
    (Select nss from empleado,dependiente
    Where nss=nsse
    Group by nss
    Having count(*) ≥ 2 ) as a
Where a.nss=empleado.nss;
```

6. Obtener el nombre de los empleados que no tienen otras personas dependientes de ellos.

```
Select nombre,apellido
From
    (Select nss from empleado
    EXCEPT
    Select nsse as nss from dependiente) as t, empleado
Where t.nss=empleado.nss;
```

7. Obtener el nombre de los jefes que tienen por lo menos una persona dependiente de ellos.

```
Select nombre,apellido
From
    ((Select nss_jefe as nss from departamento)
    INTERSECT
    (Select nsse as nss from dependiente)) as t
Natural join empleado;
```

Cuestión 2

[15 p]

Se tiene la siguiente tabla NOTAS

Nota_IA	Nota_BD	Nota_TP
8	5	NULL
10	NULL	NULL
8	6	NULL
7	5	NULL

Se desea obtener el resultado de las siguientes consultas. Justificar las respuestas.

a) `SELECT (Nota_IA+Nota_BD) FROM NOTAS`

13
NULL
14
12

Valor nulo + valor numérico = valor nulo

b) `SELECT (Nota_IA+Nota_BD+Nota_TP) FROM NOTAS`

NULL
NULL
NULL
NULL

Valor nulo + valor numérico = valor nulo

c) `SELECT MAX(Nota_IA+Nota_BD) FROM NOTAS`

14

Devuelve un campo. Los valores nulos se ignoran en la función agregada MAX

d) `SELECT MAX(Nota_IA)+MAX(Nota_BD) FROM NOTAS`

16

Devuelve la suma de los valores máximo de Nota_IA y Nota_BD. Lo null se ignoran.

e) `SELECT COUNT(Nota_TP) FROM NOTAS`

0

Se ignoran los valores nulos y se cuentan los que son distintos.

f) `SELECT AVG(Nota_TP) FROM NOTAS`

NULL

Devuele NULL ya que todas las filas son NULL y la suma es NULL

g) `SELECT AVG(Nota_TP+Nota_DB) FROM NOTAS`

NULL

Suma de todos los valores son nulos y la media es nulo

Cuestión 4

[15 p]

Una empresa dispone de un sistema para manejar los proyectos que realiza y el personal asignado a os mismos. Se usan las siguientes tablas:

PROYECTOS(cod_proy, cod_cliente, fecha_inicio, tipo)

PERSONAL(cod_func, nombre, fecha_ingreso)

TAREAS(cod_tarea, descripción, tipo)

ASIGNACION(cod_func, cod_proy, cod_tarea)

REGISTRO_HORAS(cod_func, cod_proy, fecha, cant_horas)

Se pide:

c) Resolver las siguientes consultas en SQL (sin utilizar vistas)

- 1) Obtener los códigos de los proyectos que tienen personal asignado y que en ninguna fecha fueron trabajadas más de 40 horas en él. (Notar que no necesariamente por una única persona).

```
Select cod_proy
From asignacion
Where cod_proy not in
( Selec R.cod_proy
  From registro_horas R
  Group by cod_proy, fecha
  Having sum(cant_horas) > 40);
```

- 2) Obtener los códigos de los proyectos con el total de horas realizadas en él por el personal ingresado en la empresa antes del 04/05/71

```
Select cod_pry, sum(cant_horas)
From registro_horas R, personal P
Where p.fecha_ingreso < '04-05-1971' and p.cod_func=r.cod_func
Group by cod_proy;
```

Problema 1

[20 p]

Se tiene una base de datos con las siguientes relaciones:

RESIDENCIAS(cod-res, nom-res, cod-ins, año-constr, arquitecto, num-hab)

Contiene los datos de todas las residencias y cuyos atributos son: código de la residencia, nombre de la residencia, código de la institución que la construyó, año de construcción, arquitecto que la proyectó y número de habitaciones.

INSTITUCIONES(cod-ins, nom-ins, cant-becas)

Contiene los datos de las instituciones que construyeron residencias ya sean departamentales o particulares y cuyos atributos son: código de la institución, nombre de la institución y cantidad de becas que otorga.

DEPARTAMENTALES(cod-ins, departamento, programa, encargado, tel-contacto)

Contiene los proyectos de residencias construidas por organismos departamentales y cuyos atributos son: código de la institución. Departamento, programa o proyecto, encargado del proyecto y teléfono de contacto. Las instituciones que se encuentran en esta tabla, también están en INSTITUCIONES.

PARTICULARES(cod-ins, nom-fundador, encargado, tel-contacto)

Contiene los proyectos de residencias de fundaciones particulares y cuyos atributos son: código de la institución. Departamento, programa o proyecto, encargado del proyecto y teléfono de contacto. Las instituciones que se encuentran en esta tabla, también están en INSTITUCIONES pero no en DEPARTAMENTALES.

ESTUDIANTES(DNI, nombre, depto-origen, edad)

Contiene los datos de todos los estudiantes que se han alojado en alguna residencia y cuyos atributos son: DNI del estudiante, nombre del estudiante, departamento del que viene y edad.

EST_FAC(DNI, facultad, año-curso, promedio)

Contiene en qué facultades estudia cada estudiante. Sus atributos son: DNI del estudiante, nombre de la facultad, año que está cursando y promedio de notas. Cada estudiante de esta relación

CONTRATOS(cod-res, DNI, año, becado, habitación, teléfono)

Contiene en qué residencia se ha alojado cada estudiante en cada año, ya que los contratos de alojamiento son anuales. Sus atributos son: código de residencia, DNI del estudiante, año del contrato, indicación de existencia de beca (SI ó NO), habitación que ocupa y teléfono de la habitación. Las residencias y estudiantes de esta tabla se encuentran en RESIDENCIAS y ESTUDIANTES respectivamente.

Se pide:

- c) Escribir en SQL la consulta: “Listar los códigos y nombres de las residencias construidas por la “Fundación José Fernández” que tienen ocupadas todas sus habitaciones en el 2002”.

```
SELECT R.cod-res, R.nom-res
FROM RESIDENCIAS R, INSTITUCIONES I, CONTRATOS C
WHERE R.cod-ins = I.cod-ins AND I.nom-ins = "Fundación José Fernández" AND C.cod-res = R.cod-res AND C.año=2002
GROUP BY R.cod-res, R.nom-res
HAVING count(*)=R.num-hab;
```


- d) Escribir en SQL la consulta: “Listar el código de residencia y código de institución de las residencias con más de 100 habitaciones en las que no se ha alojado ningún estudiante mayor de 25 años”

```
SELECT R.cod-res,R.cod-ins
FROM RESIDENCIAS R
WHERE R-num-hab > 100
AND R.cod-res NOT IN (
    SELECT C.cod-res
    FROM CONTRATOS C,ESTUDIANTES E
    WHERE C.DNI = E.DNI
    AND C.cod-res = R.cod-res
    AND E.edad > 25
);
```