

Solución Práctico 4 – SQL

Ejercicio 2

Considerando los siguientes esquemas relación:

- **COMPañIA** (nombreComp, ciudadComp)

Representa la información de todas las compañías, nombre de la compañía y ciudad dónde se encuentra la misma.

- **TRABAJA** (nombreEmp, nombreComp, salario)

Representa la información de la compañía en la cual trabaja cada empleado y el salario que éste percibe.

- **GERENCIA** (nombreEmp, nombreGerente)

Representa la información que relaciona a cada empleado con su gerente.

- **VIVE** (nombre, dirección, ciudad)

Representa la información de los datos personales de los empleados, nombre del empleado, dirección y ciudad donde vive.

Donde:

$$\Pi_{\text{nombreEmp}}(\text{TRABAJA}) \subseteq \Pi_{\text{nombre}}(\text{VIVE})$$

$$\Pi_{\text{nombreComp}}(\text{TRABAJA}) \subseteq \Pi_{\text{nombreComp}}(\text{COMPañIA})$$

$$\Pi_{\text{nombreEmp}}(\text{GERENCIA}) \subseteq \Pi_{\text{nombreEmp}}(\text{TRABAJA})$$

$$\Pi_{\text{nombreGerente}}(\text{GERENCIA}) \subseteq \Pi_{\text{nombreEmp}}(\text{TRABAJA})$$

Resolver las siguientes consultas en SQL:

- a. Nombre y ciudad dónde viven las personas que trabajan en la compañía de nombre INCA.

```
select t.nombreEmp, v.ciudad
from TRABAJA as t, VIVE as v
where t.nombreEmp = v.nombre and t.nombreComp = 'INCA'
```

- b. Nombre, dirección y ciudad de las personas que trabajan en la compañía de nombre HLP.SA y ganan más de \$800.000.

```
select t.nombreEmp, v.ciudad, v.direccion
from TRABAJA as t, VIVE as v
where t.nombreEmp = v.nombre and T.nombreComp = 'HLP.SA'
and t.salario > 800000
```

- c. Nombre y dirección de las personas que viven en la misma ciudad en la cual está ubicada la compañía para la cual trabaja o viven en la misma ciudad que su gerente.

```
select v.nombre, v.direccion, P.ciudad
from VIVE as v, TRABAJA as t, COMPAÑIA as c
where t.nombreEmp = v.nombre and t.nombreComp = c.nombreComp
and
(v.ciudad = c.ciudadComp --vive en la misma ciudad de la compañía
or v.ciudad in --o vive en la ciudad de su gerente
(select vG.ciudad --ciudad del gerente
from GERENCIA as g, VIVE as vG
where vG.nombre=g.nombreGerente and g.nombreEmp = v.nombre))
```

- d. Nombre de las personas que no trabajan para la compañía AMIP.

```
select t.nombreEmp
from TRABAJA as t
where t.nombreEmp not in
(select nombre_emp
from TRABAJA as t
where t.nombreComp = 'AMIP')
```

Tecnólogo en Informática – Base de Datos 1 – Solución Práctico

e. Nombre y dirección de las personas que viven en la misma ciudad que su gerente y que no viven en la ciudad de la compañía RAMF.LTDA.

```
select v1.nombre, v1.direccion, v1.ciudad
from VIVE as v1, VIVE as v2, GERENCIA as g
where v1.nombre = g.nombreEmp and g.nombreGerente = v2.nombre and
v1.ciudad = v2.ciudad
and v1.ciudad not in(select ciudadComp
                      from COMPAÑIA as c
                      where c.nombreComp = 'RAMF.LTD')
```

f. ¿Cómo deberá modificarse el esquema relacional para que una compañía pueda estar ubicada en varias ciudades? En ese nuevo esquema, encontrar los nombres de las compañías que están ubicadas en todas las ciudades donde está ubicada la compañía FARM.SA.

```
select c.nombreComp
from COMPAÑIA as c1
where not exists
    (select * From COMPAÑIA as c2 –las que no están en la ciudad que está
    FARM
     where c2.nombreComp = 'FARM.SA'
     and (c1.nombreComp, c2.ciudad) not in (select * from COMPAÑIA)
    )
```

Ejercicio 3

Debido a la desorganización reinante en una empresa de plaza, la mesa directiva de la misma decidió nombrar a un encargado de organización. Este encargado decidió automatizar el sistema de control de asignaciones de tareas a secciones y empleados de la empresa. Para esto creó una base de datos con los siguientes esquemas relación:

- **FUNCIONARIOS** (nroF, nomF, aalng)

Representa la información de todos los empleados de la empresa. Donde nroF representa el número de funcionario, nomF el nombre del funcionario y aalng representa el año de ingreso del funcionario a la empresa.

Tecnólogo en Informática – Base de Datos 1 – Solución Práctico

- SECCIONES (codS, nomS, local, tel)

Representa la información de todas las secciones que posee la empresa. Donde codS representa el código de la sección, nomS su nombre, tel número de teléfono de la sección y local el local donde esta ubicada.

En un local puede haber varias secciones.

-TAREAS (codT, nomT, duración)

Representa la información de todas las tareas de la empresa. Donde codT representa el código de la tarea, nomT su nombre y duración la duración estimada en días de la misma.

- PREVIAS (codTP, codTS)

Representa la relación de orden que existe entre las tareas.

Una tupla (p,s) de esta relación, significa que la tarea con código p debe realizarse inmediatamente antes que la tarea con código s.

$$\prod_{\text{codTP}} (\text{PREVIAS}) \subseteq \prod_{\text{codT}} (\text{TAREAS})$$

$$\prod_{\text{codTS}} (\text{PREVIAS}) \subseteq \prod_{\text{codT}} (\text{TAREAS})$$

- TRABAJA (nroF, codS, hsSem, sueldo, cargo)

Representa la cantidad de horas semanales, el cargo que tiene y el sueldo que gana cada funcionario en cada sección en que trabaja.

$$\prod_{\text{nroF}} (\text{TRABAJA}) \subseteq \prod_{\text{nroF}} (\text{FUNCIONARIOS})$$

$$\prod_{\text{codS}} (\text{TRABAJA}) \subseteq \prod_{\text{codS}} (\text{SECCIONES})$$

- ASIGNACION (nroF, codT, fAsig)

Representa las asignaciones de tareas realizadas.

Una tupla (fun, tar, fe) de esta relación representa que al funcionario de número "fun" se le ha asignado la tarea con código "tar" en la fecha "fe".

$$\Pi_{\text{nroF}}(\text{ASIGNACION}) \subseteq \Pi_{\text{nroF}}(\text{FUNCIONARIOS})$$

$$\Pi_{\text{codT}}(\text{ASIGNACION}) \subseteq \Pi_{\text{codT}}(\text{TAREAS})$$

- OBLIGACIONES (codS, codT)

Representa la relación de obligatoriedad que existe entre las secciones y algunas tareas que deben realizar.

Una tupla (sec, tar) de esta relación representa que la sección con código "sec" está obligada a realizar la tarea con código "tar".

$$\Pi_{\text{codS}}(\text{OBLIGACIONES}) \subseteq \Pi_{\text{codS}}(\text{SECCIONES})$$

$$\Pi_{\text{codT}}(\text{OBLIGACIONES}) \subseteq \Pi_{\text{codT}}(\text{TAREAS})$$

La mesa directiva de la empresa, le propone a usted que resuelva las siguientes consultas en SQL. En caso de ser necesario utilice vistas para su resolución.

a. Obtener el código, el nombre y la duración de las tareas que se deben realizar inmediatamente antes de la tarea con código "c42".

```
select codT, nomT, duracion
from TAREAS, PREVIAS
where codTS = 'c42' and codTP = codT;
```

b. Obtener los nombres y año de ingreso de los funcionarios que tienen algún sueldo mayor que algún sueldo del funcionario con número 812.

```
select nomF, aaIng
from FUNCIONARIOS
where nroF in
(select t1.nroF – funcionarios con sueldo mayor al sueldo de '812'
from TRABAJA as t1, TRABAJA as t2
where t2.nroF = '812'
and t1.sueldo > t2.sueldo)
```

d. Obtener los nombres de los funcionarios tales que en la misma fecha le asignaron más de una tarea.

```
select f.nomF
from FUNCIONARIOS as f, ASIGNACION as a
where f.nroF = a.nroF
group by f.nroF, f.nomF, a.fAsig
having count (*) > 1
```

e. Obtener los códigos, nombre y duración de las tareas que están obligadas a realizar la sección con nombre "PROMOCION" y/o la sección con nombre "COMPUTOS".

```
select t.codT, t.nomT, t.duracion
from SECCIONES as s, OBLIGACIONES as o, TAREAS as t
where s.nomS = 'PROMOCION' and s.codS = o.codS and o.codT = t.codT
union
select t.codT, t.nomT, t.duracion
from SECCIONES as s, OBLIGACIONES as o, TAREAS as t
where s.nomS = 'COMPUTOS' and s.codS = o.codS and o.codT = t.codT
```

Otra solución, con OR

```
select t.codT, t.nomT, t.duracion
from SECCIONES as s, OBLIGACIONES as o, TAREAS as t
where (s.nomS = 'PROMOCION' or s.nomS = 'COMPUTOS' )
and s.codS = o.codS and o.codT = t.codT
```

h. Hallar los nombres de los funcionarios que ingresaron antes del año 1986 y sólo fueron asignados a tareas cuya duración no supera los 60 días.

```
select f.nomF
from FUNCIONARIOS as f, ASIGNACION as a, TAREAS as t
where f.nroF = a.nroF and t.codT = a.codT and
t.duracion < 60 and f.aaIng < 1986
and not exists(select *
               from ASIGNACION as a1, TAREAS as t1
               where a1.nroF = f.nroF and a1.codT = Tt1.codT and t1.duracion >= 60)
```