

# Bases de données Relationnelles

## TP3

### MAM4- SI3

La base de données du TP3 est composée des tables décrites en Annexe

## 1 Récursivité

Cette section n'utilise que la table employees

1. Ecrire une requête qui affiche tous les couples (employee,manager) où manager est le l'identifiant d'un chef direct ou indirect de l'employé d'identifiant employee

<i>employee</i>	<i>manager</i>
101	100
102	100
103	102
103	100
...	.....
121	100
122	100
206	100
206	101
206	205

(74 rows)

---

```
with recursive hierarchie(soldat, general) as (  
select employees.employee_id , employees.manager_id  
from employees where manager_id is not null  
union  
select employees.employee_id ,hierarchie.general from  
employees join hierarchie on (employees.manager_id= hierarchie.soldat))  
select soldat as employee, general as manager from hierarchie  
order by soldat;
```

---

2. une requête qui affiche tous les couples employee , manager où manager est le nom d'un chef direct ou indirect de l'employé de nom employee

<i>employee</i>	<i>manager</i>
Kochhar	King
De Haan	King
Hunold	De Haan
Hunold	King
...	.....
Gietz	King
Gietz	Kochhar
Gietz	Higgins

(74 rows)

---

```

with recursive hierarchie(soldat, general) as (
select employees.employee_id , employees.manager_id
from employees where manager_id is not null
union
select employees.employee_id ,hierarchie.general from
employees join hierarchie on (employees.manager_id= hierarchie.soldat))

select S.last_name as employee, G.last_name as manager
from hierarchie
join employees S on(S.employee_id = hierarchie.soldat)
join employees G on (G.employee_id = hierarchie.general)
order by soldat;

```

3. Écrire une requête qui affiche tous les triplets employee , manager , distance où manager est le nom d'un chef direct ou indirect de l'employé de nom employee et distance est la distance hiérarchique entre les deux ( 1 pour un chef direct, 2 pour le chef direct du chef direct, etc....)

<i>employee</i>	<i>manager</i>	<i>distance</i>
Kochhar	King	1
De Haan	King	1
Hunold	De Haan	1
....	.....	.....
Gietz	Higgins	1
Gietz	Kochhar	2
Gietz	King	3

(74 rows)

```

with recursive hierarchie(soldat, general, distance) as (
select employees.employee_id , employees.manager_id ,1
from employees where manager_id is not null
union
select employees.employee_id ,hierarchie.general , hierarchie.distance +1 from
employees join hierarchie on (employees.manager_id= hierarchie.soldat))

select S.last_name as employee, G.last_name as manager , distance
from hierarchie
join employees S on(S.employee_id = hierarchie.soldat)
join employees G on (G.employee_id = hierarchie.general)
order by soldat, distance;

```

4. Écrire une requête qui affiche pour tous les employés leur identifiant, leur nom, leur "distance" au PDG, le nom de tous leurs chefs directs ou indirect , les identifiants de tous leur chefs directs ou indirects.

Utilisez la fonction string\_agg (consultez la doc !!) qui agrège en les concaténant des chaînes de caractères, et comme des entiers ne sont pas des chaînes de caractères vous aurez aussi besoin de caster les entiers en chaînes de caractères : id::varchar transforme l'entier id en chaîne de caractère contenant l'écriture de l'entier en base dix.

<i>Employé</i>	<i>Nom</i>	<i>niveau</i>	<i>Noms des chefs</i>	<i>Id des chefs</i>
105	Austin	3	Hunold - De Haan - King	103 102 100
110	Chen	3	Greenberg - Kochhar - King	108 101 100
104	Ernst	3	Hunold - De Haan - King	103 102 100
109	Faviet	3	Greenberg - Kochhar - King	108 101 100
206	Gietz	3	Higgins - Kochhar - King	205 101 100
107	Lorentz	3	Hunold - De Haan - King	103 102 100
106	Pataballa	3	Hunold - De Haan - King	103 102 100
113	Popp	3	Greenberg - Kochhar - King	108 101 100
111	Sciarra	3	Greenberg - Kochhar - King	108 101 100
112	Urman	3	Greenberg - Kochhar - King	108 101 100
204	Baer	2	Kochhar - King	101 100
116	Baida	2	Raphaely - King	114 100
192	Bell	2	Vollman - King	123 100
119	Colmenares	2	Raphaely - King	114 100
193	Everett	2	Vollman - King	123 100
202	Fay	2	Hartstein - King	201 100
108	Greenberg	2	Kochhar - King	101 100
205	Higgins	2	Kochhar - King	101 100
118	Himuro	2	Raphaely - King	114 100
103	Hunold	2	De Haan - King	102 100
115	Khoo	2	Raphaely - King	114 100
203	Mavris	2	Kochhar - King	101 100
126	Mikkilineni	2	Weiss - King	120 100
117	Tobias	2	Raphaely - King	114 100
200	Whalen	2	Kochhar - King	101 100
102	De Haan	1	King	100
121	Fripp	1	King	100
178	Grant	1	King	100
201	Hartstein	1	King	100
179	Johnson	1	King	100
122	Kauffing	1	King	100
101	Kochhar	1	King	100
177	Livingston	1	King	100
146	Partners	1	King	100
114	Raphaely	1	King	100
145	Russell	1	King	100
176	Taylor	1	King	100
123	Vollman	1	King	100
120	Weiss	1	King	100

(39 rows)

---

```

select employees.employee_id , employees.manager_id ,1
from employees where manager_id is not null
union
select employees.employee_id ,hierarchie.general, niveau+1 from
employees join hierarchie on (employees.manager_id= hierarchie.soldat))

select e2.employee_id as "Employé", e2.last_name as "Nom", count(general) as niv
string_agg( e1.last_name, '␣-␣' order by niveau) as "Noms␣des␣chefs",
string_agg(general::varchar, '|' order by niveau) as "Id␣des␣chefs"
from hierarchie
join employees e1 on (general=e1.employee_id)
JOIN EMPLOYEES E2 on (soldat=e2.employee_id)
group by e2.employee_id
order by niveau desc, e2.last_name

```

---