TD Fonctions II (Fonctions PL/PgSQL)

Avec solutions

- **L3 MIASHS/Ingémath**
- **Université Paris Cité**
- Année 2024-2025
- Course Homepage

- Moodle

On donne ci-dessous une requête renvoyant les paires d'acteurs ayant joué dans un même film.

```
select DISTINCT a.actor_id, b.actor_id from sakila.actor a , sakila.actor b where
exists (
select * from sakila.film_actor c where c.actor_id = a.actor_id and
exists ( select * from film_actor d where d.film_id = c.film_id and d.actor_id = b.actor_id
));
```

Pour toute paire de numéro d'acteur x, y, on pose $m(x, y)$ ssi x et y on joué dans un même film. On définit la distance d entre deux acteurs de la façon suivante. Si x est un numéro d'acteur alors $d(x, x) = 0$. Pour tout entier $k > 0$, pour toute paire d'acteurs x, y, la fonction distance obéit à la définition inductive suivante :

$$d(x, y) = k \text{ ssi } \exists z\ m(x, z) \wedge d(z, y) = k - 1.$$

Donc la distance entre deux acteurs différents est 1 s'ils ont joué dans un même film, 2 s'ils n'ont pas joué ensemble mais s'il existe un acteur qui a joué avec les deux, etc. On pose $d(x, y) = \infty$ s'il n'existe aucune façon d'aller de x à y en suivant un chemin de la relation "avoir joué ensemble dans un film".

- Écrire une fonction pl/pgsql prenant en entrée deux numéros d'acteurs et renvoyant leur distance si celle-ci est inférieure à 5 et renvoyant 5 sinon. L'en-tête de la fonction sera la suivante :

```
create or replace function nb_co_actors (act1 int,act2 int)
returns int as $plpgsql$
...
$plpgsql$ language plpgsql ;
```

- Généraliser votre fonction pour déterminer si la distance entre deux acteurs est finie ou pas.

```
-- Creer dans une table
create or replace table avigny59.co_actors as (
select DISTINCT a.actor_id actor1, b.actor_id actor2 from sakila.actor a , sakila.actor b where
exists (
select * from sakila.film_actor c where c.actor_id = a.actor_id and
exists ( select * from film_actor d where d.film_id = c.film_id and d.actor_id = b.actor_id
)
)
)
;

-- Distance si <6, 5 sinon
create or replace function avigny59.nb_co_actors (act1 int,act2 int)
```

```
returns int as
$$
Declare
c integer;
r integer;
begin
create temporary table tmp (actor1 INT , actor2 INT) on commit drop;
create temporary table tmp2 (actor1 INT , actor2 INT) on commit drop;
insert into tmp (select * from avigny59.co_actors where actor1 = act1);
c:=0;
loop
c:=c+1;
perform * from tmp where actor1 = act1 and actor2 = act2;
if found or c > 4 then
exit;
end if;
delete from tmp2;
insert into tmp2 (select distinct a.actor1, b.actor2 from tmp a join avigny59.co_actors b on a.acto
delete from tmp;
insert into tmp (select * from tmp2);
end loop;
drop table tmp;
drop table tmp2;
return c;
END ;
$$language plpgsql

-- Fonction renvoyant la distance entre un acteur et tous les autres
DECLARE
actor smallint ;
c integer;
BEGIN
FOR actor in select distinct actor_id from sakila.film_actor
LOOP
select * into c from ardurand.nb_co_actors(act1,actor);
RETURN QUERY VALUES(actor,c);
END LOOP ;
END
$$ LANGUAGE plpgsql;
```