

## TD Fonctions II (Fonctions PL/PgSQL)

- [L3 MIAHS/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  $y$  ont 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.