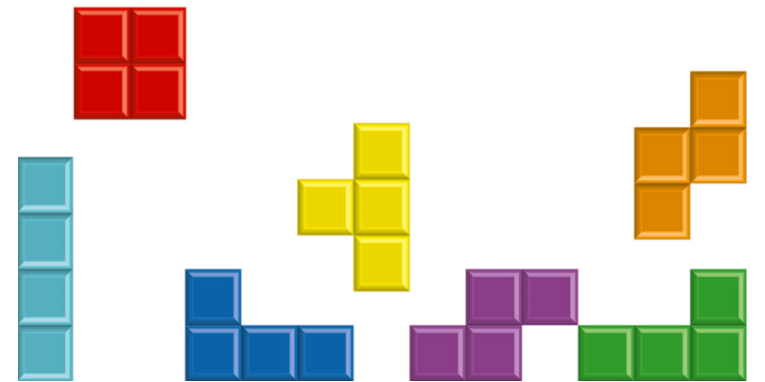


Requêtes imbriquées

Requêtes sur le résultat d'une requête



Sommaire

- Introduction
- Requêtes imbriquée retournant une table ou un champ
- Requêtes imbriquée retournant une colonne
- Requêtes imbriquées testant l'existence d'une valeur
- Requêtes imbriquées retournant une valeurs
- Les jointures entre requêtes de sélection : requêtes corrélées
 - L'union : UNION
 - L'intersection : INTERSECT
 - L'exclusion : EXCEPT/MINUS

Introduction

Dans certains cas il peut être nécessaire de réaliser une requête non pas sur une table, mais sur le résultat d'une autre requêtes : c'est la que les **requêtes imbriquées** ou **sous requêtes** entre en jeux.

- L'imbrication n'a pas vraiment de limite technique.
- L'or de requête imbriqué le premier *select* est appelé requête principale, le deuxième sous requêtes 1, et ainsi de suite.
- L'exécution s'effectue toujours de la requête la plus « profonde » dans l'imbrication vers la requête la principale.

Requêtes imbriquée retournant une table ou un champ

Ce type de requête peut être utilisé dans un **from** ou un **select**, comme source de données de la requête principale.

C'est utile dans le **from** pour limiter notamment le nombre d'enregistrements sur le quel le reste de la requête doit s'appliquer.

Pour le **select**, cela permet d'afficher dans une requête un champs provenant d'une autre requête, sans avoir à réaliser de jointure particulière.

Requêtes imbriquée retournant une table ou un champ

Exemple :

On peut calculer le nombre de film de la catégorie « Action » avec la requête suivante (même si on peut le faire avec un group by) :

```
select category.name, (  
    select count(film_category.film_id)  
    from film_category  
    join category  
    on film_category.category_id = category.category_id  
    where category.name = 'Action') as Nb_films  
from category  
where category.name = 'Action';
```

name	Nb_films
Action	64

Requête imbriquée retournant une colonne

Ce type de requête peut être utilisé avec le prédicat **IN** (ou **NOT IN**) pour l'évaluation d'un champ par rapport à une liste de valeurs retournées par cette requête.

Exemple :

On peut afficher la liste des films qui ont été loués, c'est à dire se dont on retrouve une trace dans la table *rental*.

```
select film.title from film
  where film_id in (
    select film_id from inventory
      inner join rental on inventory.inventory_id = rental.inventory_id);
```

title
ACADEMY DINOSAUR
ACE GOLDFINGER
ADAPTATION HOLES
AFFAIR PREJUDICE
AFRICAN EGG
AGENT TRUMAN
AFRO DISCO

Requêtes imbriquées testant l'existence d'une valeur

L'utilisation d'une requête imbriquée avec le prédicat **IN** ne doit être confondu avec l'utilisation du prédicat **EXISTS** (ou **NOT EXISTS**).

- Celui-ci ne vérifie pas la concordance avec une ou plusieurs valeurs, mais bien si une valeur existe ou non.
- Pour un jeu d'enregistrement équivalents, ce prédicat est donc plus performant que **IN** car il ne va pas traiter toute les possibilités, mais s'arrêter dès que l'existence est vérifier.

Requêtes imbriquées testant l'existence d'une valeur

Exemple :

On peut connaître les films ayant uniquement pour acteur référencé « MCCONAUGHEY CARY ».

```
select f.film_id, title from film f
  join film_actor on f.film_id = film_actor.film_id
  join actor on film_actor.actor_id = actor.actor_id
    and concat(last_name, ' ', first_name) = 'MCCONAUGHEY CARY'
where not exists (
  select film.film_id, title from film
    join film_actor on film.film_id = film_actor.film_id
    join actor on film_actor.actor_id = actor.actor_id
      and concat(last_name, ' ', first_name) <> 'MCCONAUGHEY CARY'
  where f.film_id = film.film_id);
```

film_id	title
240	DOLLS RAGE

Le 1^{er} select renvoie la liste des films dans les quel à joué « MCCONAUGHEY CARY »

Le 2^{ème} select renvoie la liste des films dans les quel à joué n'importe quel acteurs sauf « MCCONAUGHEY CARY »

La clause where permet de spécifier sur quel champs des deux requêtes le test doit être fait notamment en spécifiant un aliace dans le 1^{er} select.

Requêtes imbriquée retournant une valeur

Se type de requête peut être utilisé comme champ d'un SELECT de la requête principale, ou comme valeur d'une condition dans un **WHERE** ou un **HAVING**.

Exemple :

Pour afficher les films appartenant à la même catégorie que le film « DOLLS RAGE »

```
select title from film
  join film_category on film.film_id = film_category.film_id
 where category_id = (
   select category_id from film_category
   join film on film_category.film_id = film.film_id
   where title = 'DOLLS RAGE');
```

title
ANNIE IDENTITY
ARMAGEDDON LOST
ATTACKS HATE
BADMAN DAWN
BARBARELLA STREETCAR
BEVERLY OUTLAW

Les jointure entre requêtes de sélection : requête corrélées

Il est possible d'effectuer des opérations ensemblistes entre les résultats de deux requêtes de sélection ayant la même structure, c'est-à-dire ayant le même nombre de champs de type identique.

Ces opération sont :

- **UNION**
- **INTERSECT**
- **EXCEPT** (ou **MINUS** suivant les SGBD)

Travail à faire : à vous de creuser, faite une veille sur ces trois opérations.

