

Bases de Données

Partie 4 : SQL – Opérations diverses et sous-requêtes

Master 2 CGSI

CY Cergy Paris Université

Tianxiao.Liu@cyu.fr

Plan

- Mot clé **DISTINCT**
- Mise à jour des données
- Suppression des données
- Tri du résultat
- Sous-requêtes

DISTINCT

- Motivation : éviter les doublons
 - Doublons : plusieurs lignes dans le résultat ont les mêmes valeurs dans les colonnes.
 - On voudrait afficher dans le résultat une seule ligne parmi les lignes doublons.

- Syntaxe

```
SELECT DISTINCT ...
```

```
...
```

- Exemple

```
SELECT DISTINCT c.nom  
FROM client AS c;
```

Mise à jour des données 1/2

- Principe

- Retrouver les lignes que l'on souhaite mettre à jour avec la clause `WHERE`
- Mettre à jour les lignes trouvées avec de nouvelles valeurs pour les colonnes indiquées

- Syntaxe

```
UPDATE nom_table AS alias
SET  alias.colonne1 = valeur1,
     alias.colonne2 = valeur2,
     ... ..
     alias.colonneN = valeurN
WHERE condition ;
```

Mise à jour des données 2/2

- Exemples

```
UPDATE client AS c  
SET c.id_adresse = 1  
WHERE c.prenom='Paul';
```

```
UPDATE commande AS co  
SET co.quantite = co.quantite + 10  
WHERE co.id_client = 2;
```

```
UPDATE client AS c  
SET c.id_client = 2;  
WHERE c.prenom='Paul';
```

Ce n'est pas possible car id_client est la clé primaire. La requête ne peut pas s'exécuter.

Suppression des données

- Syntaxe

```
DELETE FROM nom_table AS alias  
WHERE condition;
```

- Exemple

```
DELETE FROM client AS c  
WHERE c.nom = 'Dubois';
```

Attention, si la ligne contient des valeurs utilisées comme clé étrangère, la requête ne peut pas s'exécuter.

Tri du résultat

- Syntaxe

```
SELECT ...  
FROM ...  
WHERE ...  
ORDER BY nom_colonne ASC (ou DESC);
```

- Exemple

```
SELECT *  
FROM client AS c  
ORDER BY c.nom ASC;
```

Remarque : si on ne met pas ASC ou DESC, par défaut, le système considèra que c'est ASC.

Sous-requêtes 1/4

- Motivations

- Permet de réutiliser le résultat d'une requête dans une autre requête
- **Plus lisible et compréhensible** que les requêtes jointures dans certains cas

- Syntaxe

```
SELECT ...  
FROM ...  
WHERE colonne compareur  
      (SELECT ...  
        FROM ...  
        WHERE ...)
```

La compareur peut être ">, >=, <=, <, <>, =" ou "IN"

Sous-requêtes 2/4

- Exemple 1

```
SELECT DISTINCT v.couleur
FROM voiture AS v
WHERE v.id_voiture IN
      (SELECT co.id_voiture
       FROM commande AS co
       WHERE co.quantite > 20);
```

Cette requête liste les couleurs des voitures qui ont été commandées en grande quantité (plus de 20).

- **Réflexion : peut-on répondre à la même question en réalisant une autre requête sans sous-requête?**

Sous-requêtes 3/4

- Exemple 2

```
SELECT v.marque
FROM commande AS co INNER JOIN voiture AS v
      ON (co.id_voiture = v.id_voiture)
WHERE co.quantite >
      (SELECT AVG(com.quantite)
       FROM commande as com);
```

Cette requête retrouve les marques des voitures qui ont été commandées avec une quantité supérieure à la moyenne des quantités de toutes les commandes.

Sous-requêtes 4/4

- Exemple 3

```
SELECT c.nom
FROM client AS c INNER JOIN commande AS co
      ON (c.id_client = co.id_client)
WHERE co.id_voiture =
      (SELECT v.id_voiture
       FROM voiture AS v
       WHERE v.marque = 'Renault')
AND co.quantite >= 10;
```

Cette requête retrouve le nom des clients ayant commandé des voitures "Renault" avec une quantité supérieure ou égale à 10.