

Bases de Données Relationnelles

Partie 2 : SQL - Sélection et jointure interne

Master 2 CGSI

CY Cergy Paris Université

Tianxiao.Liu@cyu.fr

Plan : séance 2

- Sélection (recherche) et projection des données
 - SQL – SELECT ... FROM...
 - SQL – SELECT ... FROM ... WHERE ...

- Jointure interne
 - Motivation et mécanisme
 - Jointure entre deux tables
 - Jointures entre plusieurs tables (> 2)

Sélection et projection 1/3

- Objectif
 - Retrouver les lignes dans une table satisfaisant à des critères (conditions) de recherche
- Solution
 - Requêtes SQL
 - Ce que l'on doit préciser dans ces requêtes
 - Dans quelle table la recherche s'effectue ?
 - Format du résultat (quelles colonnes ?) → Projection (**projeter ce que l'on veut**)
 - Critères de sélection ?

Sélection et projection 2/3

- Syntaxe

```
SELECT alias.colonne1, alias.colonne2, ...  
FROM nom_table AS alias;
```

- Alias

- Un autre nom (normalement plus court) de la table
- Unique référence dans toute la requête

- Symbole spécifique

- * : Si l'on veut projeter toutes les colonnes de la table

Sélection et projection 3/3

- Exemples

```
SELECT c.nom, c.prenom  
FROM client AS c;
```

```
SELECT a.numero, a.rue, a.ville  
FROM adresse AS a;
```

```
SELECT *  
FROM client AS c;
```

Prédicat de sélection 1/6

- Sélection avec critères
 - Objectif : ne sélectionner que les données satisfaisant à des conditions
- Exprimer les critères
 - En utilisant des **prédicats de sélection**
 - Ce sont des **expressions logiques** : vrai ou faux
 - Combinaison des expressions
 - Satisfaire en même temps à plusieurs conditions : **AND**
 - Satisfaire à une des conditions : **OR**
 - Utiliser les parenthèses en cas complexe (≥ 3 conditions)

Prédicat de sélection 2/6

- Une expression logique sans combinaison
 - Syntaxe : `colonne1` `comparateur` `colonne2`
- Comparateurs
 - > supérieur à
 - >= supérieur ou égal à
 - < inférieur à
 - <= inférieur ou égal à
 - = égal à
 - <> différent de

Prédicat de sélection 3/6

■ Exemples

`c.nom='Jean'`

➤ `c` est l'alias de la table `client`

`a.ville='Paris'`

➤ `a` est l'alias de la table `adresse`

`c.date_naissance < '1980-08-02'`

➤ Les clients nés avant le 2 août 1980

`c.date_naissance < '1980-08-02' AND c.nom='Dupont'`

➤ Les clients nés avant le 2 août 1980 ayant nom de famille 'Dupont'

`c.date_naissance > '1980-08-02' OR c.sexe = FALSE`

➤ Les clients féminins **ou** les clients nés après le 2 août 1980

Prédicat de sélection 4/6

- Entre deux valeurs (incluses)

- Syntaxe : `BETWEEN valeur1 AND valeur2`

- Exemple

- ```
c.date_naissance BETWEEN '1980-08-01' AND '1985-09-01'
```

- Parmi les valeurs

- Syntaxe : `IN (valeur1, valeur2, ...)`

- Exemple

- ```
c.prenom IN ('Paul', 'Jean', 'Luc')
```

Prédicat de sélection 5/6

- Expression LIKE
 - Permet d'effectuer une recherche avec ambiguïté
 - Syntaxe : `LIKE valeur`
 - où `valeur` contient des caractères de remplacement
 - Caractères de remplacement
 - `%` : n'importe quels caractères (**0~N** caractères)
 - Exemple :

```
SELECT *
```

```
FROM client AS c
```

```
WHERE c.prenom LIKE 'Paul%';
```

Requêtes avec prédicats 1/3

- Syntaxe

```
SELECT alias.colonne1, alias.colonne2, ...  
FROM nom_table AS alias  
WHERE condition
```

- Condition

- Un prédicat de sélection
- Combinaison possible avec **AND** et **OR**

Requêtes avec prédicats 2/3

- Exemples

```
SELECT *  
FROM client AS c  
WHERE c.prenom = 'Paul';
```

➤ Retrouver tous les clients ayant le prénom 'Paul'

```
SELECT c.prenom, c.nom  
FROM client AS c  
WHERE c.prenom = 'Jean' AND  
       c.date_naissance < '1980-08-01';
```

➤ Retrouver tous les clients ayant le prénom 'Jean', nés avant le 1^{er} août 1980

Requêtes avec prédicats 3/3

- Exemples

```
SELECT *  
FROM adresse AS a  
WHERE a.rue LIKE 'rue%' AND a.ville='Paris'
```

➤ Retrouver toutes les adresses nommées 'rue ...' à Paris

```
SELECT *  
FROM client AS c  
WHERE (c.prenom='Jean' OR c.nom='Dubois') AND  
       c.date_naissance > '1980-08-01' ;
```

➤ Retrouver tous les clients nés après le 1^{er} août 1980, possédant un prénom 'Jean' **ou** un nom de famille 'Dubois'

Jointures

- Objectif
 - Relier les données **pertinentes** de différentes tables
 - Les données ont des liens logiques entre elles
 - références, clés étrangères, même type de données...
- Exemple
 - Retrouver les informations des clients et les informations de l'adresse de chacun de ces clients
 - Il s'agit d'une sélection des données de deux tables
 - Joindre les lignes des deux tables ensemble

Jointures internes 1/4

- Principe
 - Prédicat de jointure
 - Pour comparer les colonnes dans les deux tables mentionnées dans le prédicat (souvent **clé étrangère**)
 - Procédure
 - Pour chaque ligne de la première table, on vérifie toutes les lignes de la deuxième table pour trouver toutes les combinaisons satisfaisant au prédicat de jointure
 - Résultat de la jointure (pas forcément de la requête)
 - Les lignes dans le résultat contiennent toutes les colonnes des deux tables

Jointures internes 2/4

- Syntaxe

```
SELECT ...
```

```
FROM (nom_table1 AS alias1
```

```
    INNER JOIN nom_table2 AS alias2
```

```
    ON (alias1.colonne1 = alias2.colonne2))
```

```
    INNER JOIN nom_table3 AS alias3
```

```
    ON (alias2.colonne3 = alias3.colonne4)
```

```
    ... ..
```

- Résumé

- Les jointures peuvent être réalisées entre plusieurs tables, il faut utiliser les **parenthèses** si nombre de tables > 2 .

Jointures internes 3/4

- Exemples

```
SELECT *  
FROM client AS c INNER JOIN adresse AS a  
      ON (c.id_adresse = a.id_adresse);
```

```
SELECT c.nom, a.ville  
FROM client AS c INNER JOIN adresse AS a  
      ON (c.id_adresse = a.id_adresse);
```

```
SELECT c.nom, a.ville  
FROM client AS c INNER JOIN adresse AS a  
      ON (c.id_adresse = a.id_adresse)  
WHERE a.ville = 'Paris';
```

Jointures internes 4/4

- Exemples

```
SELECT *
FROM (client AS c INNER JOIN adresse AS a
      ON (c.id_adresse = a.id_adresse))
     INNER JOIN commande AS co
      ON (c.id_client = co.id_client);
```

```
SELECT c.nom, c.prenom, a.ville
FROM (client AS c INNER JOIN adresse AS a
      ON (c.id_adresse = a.id_adresse))
     INNER JOIN commande AS co
      ON (c.id_client = co.id_client)
WHERE a.ville = 'Paris' AND co.quantite = 10;
```