

## 6. SQL — consultation

### Les requêtes de consultation

- Représente la majorité des requêtes
- Encapsule complètement l'algèbre relationnel
- Une seule commande !

136

### Les requêtes de consultation

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]  
* | expression [ AS output_name ] [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE condition ]  
[ GROUP BY expression [, ...] ]  
[ HAVING condition [, ...] ]  
[ { UNION | INTERSECT | EXCEPT [ ALL ] } select ] [ ORDER BY  
expression [ ASC | DESC | USING operator ] [, ...] ]  
[ FOR UPDATE [ OF tablename [, ...] ] ]  
[ LIMIT { count | ALL } [ { OFFSET | , } start ] ]
```

137

### Les requêtes de consultation

```
SELECT * FROM auteur;
```

```
num_a |      nom  
-----+-----  
    1 | Albert Uderzo  
    2 | Victor Hugo  
    3 | J.K. Rowling  
(3 lignes)
```

```
Temps : 0,526 ms
```

138

## Les requêtes de consultation

```
SELECT * FROM auteur;

num_a | nom
-----+-----
1 | Albert Uderzo
2 | Victor Hugo
3 | J.K. Rowling
(3 lignes)

Temps : 0,526 ms
```

139

---

---

---

---

---

---

---

---

## Les requêtes de consultation

```
SELECT * FROM livre;

num_l | titre | auteur
-----+-----+-----
1 | Le fils d Asterix | 1
4 | Harry Potter à l ecole des sorciers | 3
5 | Harry Potter et la chambre des secrets | 3
2 | Les miserables | 2
3 | Notre dame de Paris | 2
(5 lignes)

Temps : 0,286 ms
```

140

---

---

---

---

---

---

---

---

## Les requêtes de consultation

```
SELECT * FROM editeur;

num_e | nom | ville
-----+-----+-----
1 | Albert-Rene | Bruxelles
2 | Gallimard | Paris
3 | Folio | Paris
(3 lignes)

Temps : 0,239 ms
```

141

---

---

---

---

---

---

---

---

## Les requêtes de consultation

```
SELECT * FROM edite_par;

num_l | num_e | date
-----+-----+-----
1 | 1 | 1998-03-24
4 | 2 | 1999-03-01
5 | 2 | 2000-02-01
2 | 3 | 1940-02-02
3 | 2 | 1967-06-12
(5 lignes)

Temps : 0,239 ms
```

142

---

---

---

---

---

---

---

---

## Les requêtes de consultation

```
SELECT * FROM utilisateur;
```

num_u	nom	prenom
1	Liya	Fritz
2	Shamas	Lake
3	Adelaide	Khan
4	Amisha	Sheridan

(4 lignes)

Temps : 0,230 ms

143

---

---

---

---

---

---

---

---

## Les requêtes de consultation

```
SELECT * FROM emprunt;
```

num_l	num_u
1	1
2	4
4	1

(3 lignes)

Temps : 0,248 ms

144

---

---

---

---

---

---

---

---

## Les requêtes de consultation

```
SELECT * FROM reserve;
```

num_l	num_u
1	2
4	2

(2 lignes)

Temps : 0,266 ms

145

---

---

---

---

---

---

---

---

## Consultation simple

- ▶ Syntaxe:
  - ▶ `SELECT col1, col2, ..., coln FROM nomTable;`
  - ▶ Variante usuelle: `SELECT * FROM nomTable;`
  - ▶ Exemple: `SELECT * FROM utilisateur ;`

146

---

---

---

---

---

---

---

---

## Expression d'une projection

- ▶ **Définition** : la projection d'une relation R de schéma  $R(A_1, A_2, \dots, A_n)$  sur les attributs  $A_{i_1}, A_{i_2}, \dots, A_{i_p}$  ( $p < n$ ) est une relation  $R_0(A_{i_1}, A_{i_2}, \dots, A_{i_p})$  dont les tuples sont obtenus par élimination des valeurs des attributs de R n'appartenant pas à  $R_0$ .
- ▶ Syntaxe: `SELECT coli1, coli2, ..., colip FROM table_name;`
- ▶ Exemple: `SELECT nom, prenom FROM utilisateur ;`
- ▶ On peut inverser l'ordre de présentation (aucun impact sur le calcul)
- ▶ La clause **DISTINCT** (permet de supprimer les doublons)

147

## Restriction

- ▶ **Définition** : la restriction (ou sélection) de la relation R par une qualification Q est une relation  $R_0$  de même schéma dont les tuples sont ceux de R satisfaisant la qualification Q.
- ▶ La qualification peut être exprimée à l'aide de constantes, comparateurs arithmétiques, opérateurs logiques
- ▶ Prédicat:
  - ▶ La qualification est de la forme `<attribut> <opérateur> <valeur>` avec opérateur { =, !=, <, <=, >, >= }
  - ▶ Il est possible de composer plusieurs conditions de base à l'aide des opérateurs booléens de disjonction (OR), conjonction (AND), négation (NOT).

148

## Expression d'une Restriction

- ▶ Introduction clause **WHERE**
- ▶ Utilisation des opérateurs booléens : **AND, OR, NOT**
- ▶ Comparaison de chaînes, dates, d'entiers, ...
- ▶ Exemple:  
`SELECT * FROM livre WHERE auteur=2 ;`

149

## Traitement des chaînes

- ▶ Opérateur **LIKE**
- ▶ Caractère spéciaux : **'%'** (remplace de 0 à plusieurs caractères) et **'?'** (remplace exactement un caractère).
- ▶ Exemple:  
`SELECT DISTINCT titre FROM livre WHERE titre LIKE 'H%' ;`

150

## Traitement des chaînes

- ▶ Opérateur de comparaison =, <>, >, <, >=, <=, ... (ordre lexicographique) (aussi applicable à tout type **INTEGER**, **DATE**...)
- ▶ Opérateur de concaténation ||, fonctions prédéfinies (ex : **UPPER**, **LOWER**)
- ▶ Exemple:  

```
SELECT UPPER(nom || ' ' || prenom) AS nom_prenom  
FROM utilisateur;
```

Renommage d'attribut

151

---

---

---

---

---

---

---

---

## Traitement des chaînes

- ▶ Comparaison de chaînes : clause **BETWEEN** permet de vérifier si la valeur d'un attribut est comprise entre deux constantes
- ▶ Exemple :  

```
select nom from utilisateur where nom between  
'A%' and 'F%' ;
```
- ▶ Note : l'exemple suivant est identique :  

```
select nom from utilisateur where nom >= 'A%'  
and nom <= 'F%' ;
```
- ▶ Applicable à tout type (entier, chaîne, date, ...)

152

---

---

---

---

---

---

---

---

## Présentation des données

- ▶ Ordre d'affichage des colonnes
- ▶ Clause **DISTINCT**, évite les doublons
- ▶ Ordre d'affichage des lignes, clause **ORDER BY**
- ▶ Ordre des lignes multi-critères
- ▶ *Aucun impact sur le traitement algébrique des requêtes*

153

---

---

---

---

---

---

---

---

## Opérations de calcul

- ▶ Opérateurs arithmétiques : +, -, ...
- ▶ Exemple :  

```
SELECT NOW()-date_edition AS duree, num_l FROM  
edite_par ;
```
- ▶ Expressions arithmétiques applicables dans la clause **WHERE**

154

---

---

---

---

---

---

---

---

## Fonctions de calcul

- ▶ Une fonction de calcul est une fonction qui s'applique sur un ensemble de tuples et qui renvoie une valeur unique
- ▶ Syntaxe:  
`nomFonction(nomColonne)` ou `nomFonction(*)`
- ▶ Résultat est stocké dans une colonne correspondant au nom de la fonction.
- ▶ Toujours une ligne résultat.
- ▶ Fonctions standards : `COUNT`, `MIN`, `MAX`, `AVG`, `SUM`

155

---

---

---

---

---

---

---

---

## Fonctions de calcul — exemple

- ▶ Exemple avec renommage :  
`SELECT COUNT(*) AS nombre FROM livre ;`
- ▶ Exemple sans renommage :  
`SELECT MIN(num_l), MAX(num_l),  
AVG(num_l),SUM(num_l) FROM livre ;`

156

---

---

---

---

---

---

---

---

## Calcul sur des groupes de lignes

- ▶ Sélectionner des lignes pour appliquer un calcul Introduction clause `GROUP BY`
- ▶ permet de partitionner la relation en sous-relations ayant les mêmes valeurs sur les attributs précisés : on peut alors appliquer des fonctions à chaque sous- relation.
- ▶ Exemple:  
`select auteur, count(*) as nbre_par_auteur from  
livre group by auteur ;`

157

---

---

---

---

---

---

---

---

## Calcul sur des groupes de lignes

- ▶ Imposer une condition aux groupes formés par la clause `GROUP BY`
- ▶ Introduction clause `HAVING`
- ▶ Poser une condition portant sur chacune des sous-relations générées par le `GROUP BY`. Les sous-relations ne vérifiant pas la condition sont écartées du résultat.
- ▶ Exemple :  
`SELECT auteur, COUNT(*) AS nbre_par_auteur FROM  
livre GROUP BY auteur HAVING COUNT(*)>1 ;`

158

---

---

---

---

---

---

---

---

## Produit Cartésien

- ▶ Définition : Le produit cartésien de deux relations R et S est une relation T ayant pour schéma la concaténation de celui de R avec celui de S et pour tuples toutes les combinaisons des tuples de R et S.
- ▶ Opérateur commutatif :  $R \times S = S \times R$
- ▶ Opérateur intermédiaire (pas de sens en soi)
- ▶

159

---

---

---

---

---

---

---

---

## Expression Produit Cartésien

- ▶ Utilisation clause **FROM**
- ▶ Déclaration de variables (ou utiliser le nom de la table)
- ▶ Exemples :
  - ▶ `select * from utilisateur, livre ;`
  - ▶ `select distinct e.nom as nomEditeur , a.nom as NomAuteur from editeur e, auteur a;`

160

---

---

---

---

---

---

---

---

## Jointure

- ▶ **Définition** : La jointure de deux relations R et S selon une qualification multi-attributs Q est l'ensemble des tuples du produit cartésien  $R \times S$  satisfaisant la qualification Q
- ▶ Opérateur commutatif :  $R \bowtie_{\text{prédicat}} S = S \bowtie_{\text{prédicat}} R$
- ▶ Prédicat :  $\langle \text{attribut1} \rangle \langle \text{opérateur} \rangle \langle \text{attribut2} \rangle$  où attribut1 appartient à R et attribut2 appartient à S avec opérateur  $\{ =, \neq, <, <=, >, >= \}$
- ▶ Il est possible de composer plusieurs conditions de base à l'aide des opérateurs booléens de disjonction (**OR**), conjonction (**AND**), négation (**NOT**).
- ▶

161

---

---

---

---

---

---

---

---

## Expression d'une Jointure

- ▶ Relier avec cohérence plusieurs tables.
- ▶ Relier les **clés étrangères** avec les **clés primaires** correspondantes
- ▶ Exemple :

```
select titre, nom from  where  ;
```

Produit Cartésien

Prédicat de jointure

162

---

---

---

---

---

---

---

---

## Jointure — exemples

- ▶ Exemple : liste de couples de livres ayant le même auteur
- ▶ 

```
select l1.titre, l2.titre from livre l1, livre
l2 where l1.auteur=l2.auteur and l1.titre >
l2.titre ;
```
- ▶

163

---

---

---

---

---

---

---

---

## Les jointures à la SQL/2

- ▶ Nouvelles possibilités d'expression de jointures
- ▶ Non encore implémenté par tous les SGBD (ex. Oracle)
- ▶ Les expressions de jointures sont exprimés dans la clause **FROM**
- ▶ Distinction de jointures : **inner join** (défaut), **left outer join**, **right outer join**, **full outer join**
- ▶ Pour les exemples suivants :
  - ▶ 

```
insert into livre values (6, 'Le livre
inconnu', null) ;
```
  - ▶ 

```
insert into auteur values (4, 'Paltoquet') ;
```

164

---

---

---

---

---

---

---

---

## Jointure SQL/2 classique

- ▶ 

```
select titre, nom
from livre
inner join auteur
on livre.auteur=auteur.num_a ;
```
- ▶

165

---

---

---

---

---

---

---

---

## Jointure SQL/2 ext gauche

- ▶ 

```
select titre, nom
from livre
left outer join auteur
on livre.auteur=auteur.num_a ;
```
- ▶

166

---

---

---

---

---

---

---

---



## Jointure SQL/2 ext droite

- ▶ 

```
select titre, nom
from livre
right outer join auteur
on livre.auteur=auteur.num_a ;
```
- ▶

167

---

---

---

---

---

---

---

---

## Jointure SQL/2 ext complète

- ▶ 

```
select titre, nom
from livre
full outer join auteur
on livre.auteur=auteur.num_a ;
```

168

---

---

---

---

---

---

---

---

## Syntaxe des jointures

- ▶ INNER et OUTER sont toujours facultatifs
- ▶ LEFT, RIGHT et FULL impliquent une jointure externe
- ▶ Les syntaxes :
  - ▶ T1 {[INNER]([LEFT|RIGHT|FULL][OUTER])} join T2 on boolean\_expression
  - ▶ T1 {[INNER]([LEFT|RIGHT|FULL][OUTER])} join T2 using liste\_nom\_colonne
    - ▶ USING a équivalent à on t1.a = t2.a
    - ▶ USING (a,b) équivalent à on t1.a=t2.a and t1.b=t2.b
  - ▶ T1 NATURAL { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } join T2
- ▶ NATURAL effectue une comparaison de toutes les colonnes de même nom dans les deux tables.

169

---

---

---

---

---

---

---

---

## Expression d'une union

- ▶ Définition : l'union de 2 relations R et S de même schéma est une relation T de même schéma contenant l'ensemble des tuples appartenant à R ou S
- ▶ Syntaxe :
  - < requête > UNION < requête >
  - < requête > UNION ALL < requête >
- ▶ Le résultats des deux requêtes doivent avoir la même structure : même nombre de colonnes, mêmes types de données, même ordre.
- ▶ Exemple :

```
select nom from auteur union select nom from
editeur ;
```

170

---

---

---

---

---

---

---

---

## Expression d'une intersection

- ▶ Définition : L'intersection de deux relations R et S de même schéma est une relation T de même schéma contenant les tuples appartenant à la fois à R et S
- ▶ Syntaxe :  
< requête > INTERSECT < requête >
- ▶ Exemple :  
`select nom from auteur intersect select nom from editeur ;`
- ▶ est équivalent à :  
`select nom from auteur a where a.nom IN select nom from editeur ;`  
`select nom from auteur a where EXISTS select nom from editeur e where a.nom = e.nom;`
- ▶ Le résultats des deux requêtes doivent avoir la même structure
- ▶ Attention à la structure du **IN** et du **EXISTS**

## Expression d'une différence

- ▶ Définition : la différence de 2 relations R et S de même schéma est une relation T de même schéma contenant l'ensemble des tuples appartenant à R et n'appartenant pas à S
- ▶ Syntaxe :  
< requête > EXCEPT < requête >
- ▶ Exemple :  
`select nom from auteur except select nom from editeur ;`
- ▶ est équivalent à :  
`select nom from auteur a where a.nom NOT IN select nom from editeur`  
`select nom from auteur a where NOT EXISTS select nom from editeur e where a.nom = e.nom`
- ▶ Note : Attention à la structure du **NOT IN** et du **NOT EXISTS**

## Exercices

- ▶ a) Propriétaire (NumProprio, nom, prénom, adresseProprio)
- ▶ b) Pharmacie (NumPharma, nomPharma, NumProprio#, adresse, téléphone)
- ▶ c) Médicament (IdMédicament, NomMedic, marque, Type, Vignette, PrixConseillé)
- ▶ d) Stock (NumPharma#, IdMédicament#, QuantitéPossédée, PrixPratiqué)
- ▶ e) Prescription (IdMédicament#, NumPatient#, DateOrdonnance, NumPharma#, QuantitéFournie)
- ▶ f) Ordonnance (NumPatient#, DateOrdonnance, Médecin)
- ▶ g) Patient (NumPatient, NomPatient, AdressePatient, TelPatient)

## Exercices

1. Liste des noms des pharmacies
2. Liste des médicaments (tous les attributs) en vente libre (type)
3. Liste des noms des médicaments de vignettes bleues
4. Liste des noms des médicaments de vignettes bleues ou en vente libre
5. Liste des noms des médicaments en vente libre et de marque Bayer
6. Noms des pharmacies de Monsieur Durand
7. Noms des patients à qui on a prescrit du paracétamol (nomMedic)
8. Noms des patients ayant obtenus une ordonnance du Dr Maboul le 11 octobre 2010

## Exercices

9. Noms des patients à qui le Dr Maboul a prescrit du Prozac le 11 octobre 2010
10. Liste des noms des médicaments de vignettes bleues et de vignettes blanches
11. Liste des noms des médicaments qui ne sont plus en stock dans (aucune des pharmacies du) le réseau
12. Noms et prénoms des propriétaires qui ne possèdent plus de pharmacie
13. Noms et prénoms des propriétaires qui possèdent plusieurs pharmacies
14. Liste des médicaments et des pharmacies où le prix pratiqué du médicament est inférieur au prix conseillé
15. Liste des propriétaires (nom et prénom) et de leur pharmacies par ordre alphabétique des propriétaires et inverse pour les pharmacies

175

---

---

---

---

---

---

---

---

## Exercices

16. Liste des noms et prénoms (concaténé) des propriétaires ayant plus de 3 pharmacies
17. Prix moyen (forcément prix pratiqué) du paracétamol
18. Date de la dernière ordonnance de Monsieur Dupont
19. Nom des médicaments dont la moyenne des prix pratiqués est supérieure au prix conseillé
20. Coût de l'ordonnance du 08/11/2020 de Monsieur Dupont
21. Nombre de pharmacies possédées par propriétaire

176

---

---

---

---

---

---

---

---