

Introduction à l'interrogation de BD avec SQL

Dr N. BAME

Plan

- Présentation du langage SQL
- Projection et sélection
- Tri des résultats
- Opérations ensemblistes

Interrogation : syntaxe simplifiée de SQL

```
SELECT col1, ...colk  
FROM Table  
[WHERE condition] ;
```

Retourne

- col₁, ...col_k : des attributs de la table **Table** (projection)
- Des enregistrements de la table **Table**
- qui vérifient **condition** (sélection)
- La clause **WHERE** est facultative mais très utile

Projection : exemples

Schéma de la BD

Emp (Eno, Ename, Title, City)

Project(Pno, Pname, Budget, City)

Pay(Title, Salary)

Works(Eno, Pno, Resp, Dur)

- Noms de tous les employés ?
- Noms et budgets des projets ?

Projection : exemples

Schéma de la BD

Emp (Eno, Ename, Title, City)

Project(Pno, Pname, Budget, City)

Pay(Title, Salary)

Works(Eno, Pno, Resp, Dur)

- **Noms de tous les employés**

```
SELECT Ename
```

```
FROM Emp ;
```

- **Noms et budgets des projets**

```
SELECT Pname, Budget
```

```
FROM Project ;
```

Projection sur toutes les colonnes de la table

- Pour retourner **toutes les colonnes**
 - **SELECT ***

Exemple

- Toutes les informations sur les employés

Projection sur toutes les colonnes de la table

- Pour retourner toutes les colonnes
 - `SELECT *`

Exemple

- Toutes les informations sur les employés
`SELECT *`
`FROM Emp ;`

Projection avec élimination des doublons

- Sémantique « multi-ensembliste »:
 - Possibilité **d'avoir des doublons**
parce que les éliminer coûte cher, parce qu'on peut vouloir les compter,...
 - Les **éliminer** avec le mot clé **DISTINCT**
SELECT DISTINCT

Exemple

- Toutes les villes où vivent des employés

```
SELECT City  
FROM Emp ;
```
- L'ensemble des villes où vivent des employés

```
SELECT DISTINCT City  
FROM Emp ;
```


Projection : éliminer les doublons

- Sémantique « multi-ensembliste »:
 - Possibilité **d'avoir des doublons**
parce que les éliminer coûte cher, parce qu'on peut vouloir les compter,...
 - Les **éliminer** avec le mot clé **DISTINCT**
select DISTINCT

Exemple

- Toutes les villes où vivent des employés
SELECT City
FROM Emp ;
- L'ensemble des villes où vivent des employés
SELECT **DISTINCT** City
FROM Emp ;

Expressions dans les projection/sélection

- Possibilité d'exprimer des **opérations arithmétiques**
 - (att1+att2, att*1.5, etc)

- Salaires mensuel par titre (considérer que Salary est pour un an)

```
SELECT Title, Salary/12  
FROM Pay ;
```

- Titre dont le double du salaire est inférieur à 10000000

```
SELECT Title  
FROM Pay  
WHERE (salary*2) < 3000000;
```

Expressions dans les projection/sélection

- Possibilité d'exprimer des opérations arithmétiques
 - (att1+att2, att*1.5, etc)

- Salaires mensuel par titre (considérer que Salary est pour un an)

```
SELECT Title, Salary/12
```

```
FROM Pay ;
```

- Titre dont le double du salaire est inférieur à 10000000

```
SELECT Title
```

```
FROM Pay
```

```
WHERE (salary*2) < 10000000;
```

Renommage de colonne

- Possibilité de **renommer** une **colonne** dans le **SELECT** avec le mot-clé **AS**
 - Lisibilité des résultats

Exemple

- Noms et budgets des projets

```
SELECT Pname AS NomProjet, Budget AS BudgetProjet  
FROM Project ;
```

Renommage de colonne

- Possibilité de **renommer** une **colonne** dans le **SELECT** avec le mot-clé **AS**
 - Lisibilité des résultats

Exemple

- Noms et budgets des projets
SELECT Pname **AS** NomProjet, Budget **AS** BudgetProjet
FROM Project ;

Example

PROJ

| <u>PNO</u> | PNAME | BUDGET |
|------------|-------------------|--------|
| P1 | Instrumentation | 150000 |
| P2 | Database Develop. | 135000 |
| P3 | <i>CAD/CAM</i> | 250000 |
| P4 | Maintenance | 310000 |
| P5 | <i>CAD/CAM</i> | 500000 |

SELECT PNO, BUDGET
FROM PROJ :

| <u>PNO</u> | BUDGET |
|------------|--------|
| P1 | 150000 |
| P2 | 135000 |
| P3 | 250000 |
| P4 | 310000 |
| P5 | 500000 |

SELECT PNAME FROM
PROJ :

| PNAME |
|-------------------|
| Database Develop. |
| Instrumentation |
| <i>CAD/CAM</i> |
| Maintenance |
| <i>CAD/CAM</i> |

SELECT DISTINCT PNAME
FROM PROJ :

| PNAME |
|-------------------|
| Maintenance |
| <i>CAD/CAM</i> |
| Database Develop. |
| Instrumentation |

Sélection : WHERE Prédicat

Prédicats **simples** :

Expression1 θ *Expression2*

où

- *Expression1* peut être un attribut ou une expression arithmétique
- $\theta \in \{<, >, =, <=, >=, <>\}$
- et *Expression2* une expression ou une valeur de domaine

• Exemples :

Name = 'J. Doe'

(Age + 30) >= 65

R.A = S.B

Prédicats **composés** :

- prédicats simples combinés avec les connecteurs logiques :

AND, OR, NOT

Exemple de sélection

EMP

| ENO | ENAME | TITLE |
|-----|-----------|-------------|
| E1 | J. Doe | Elect. Eng. |
| E2 | M. Smith | Syst. Anal. |
| E3 | A. Lee | Mech. Eng. |
| E4 | J. Miller | Programmer |
| E5 | B. Casey | Syst. Anal. |
| E6 | L. Chu | Elect. Eng. |
| E7 | R. Davis | Mech. Eng. |
| E8 | J. Jones | Syst. Anal. |

```
SELECT * FROM EMP  
WHERE TITLE = 'Elect. Eng.'
```

| ENO | ENAME | TITLE |
|-----|--------|-------------|
| E1 | J. Doe | Elect. Eng |
| E6 | L. Chu | Elect. Eng. |

Requêtes avec prédicats

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Professions qui gagnent plus de 5000000 par an ?
- Numéros des managers d'un projet pendant plus de 17 mois?

Requêtes avec prédicats

Emp(Eno, Ename, #Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Professions qui gagnent plus de 5000000 par an ?

```
SELECT x.Title  
FROM Pay x  
WHERE x.Salary > 5000000
```

- Numéros des managers d'un projet pendant plus de 17 mois?

```
SELECT Eno  
FROM Works  
WHERE Dur > 17 AND Resp='Manager'
```

IN, BETWEEN, LIKE

- Appartenance à un ensemble de valeurs :
Att **IN** (*Const1*, *Const2*, ...)
- Appartenance à un intervalle de valeurs :
Att **BETWEEN** *Constante1* **AND** *Constante2*
- Ressemblance à un motif :
Att **LIKE** '*MOTIF*'
- où *MOTIF* combine des chaînes et des joker
 - % pour une chaîne quelconque (y compris vide)
 - _ pour un caractère quelconque et un seul

Requêtes avec prédicats (2)

Emp(Eno, Ename, #Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Nom des projets de Dakar, Thies ou Kaolack?
- Nom des projets ayant un budget compris entre 5M et 10 M?

Requêtes avec prédicats (2)

Emp(Eno, Ename, #Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Nom des projets de Dakar, Thies ou Kaolack?

```
SELECT Pname
```

```
FROM Project
```

```
WHERE City IN ('Dakar', 'Thies', 'Kaolack')
```

Comment le faire sans IN ?

- Nom des projets ayant un budget compris entre 5M et 10M?

```
SELECT Pname
```

```
FROM Project
```

```
WHERE Budget BETWEEN 5 AND 10
```

Comment le faire sans BETWEEN ?

Requêtes avec prédicats (3)

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Nom des employés commençant pas C?
- Nom des employés dont le 2ème numéro est un 5?

Requêtes avec prédicats (3)

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Nom des employés commençant pas C?

```
SELECT Ename
FROM Emp
WHERE Ename LIKE 'C%'
```

- Nom des employés dont le 2ème numéro est un 5?

```
SELECT Ename
FROM Emp
WHERE Eno LIKE '_5%'
```

Nom des employés habitant une ville composé de 2 mots (ex : Keur Massar)?

Valeurs nulles

- ❖ La valeur de certains attributs peut
 - ne pas être connue (ex. : année de construction du Louvre)
 - ou ne pas avoir de sens (ex. : nom de jeune fille pour un homme)

on parle alors de valeurs nulles (mot-clé **NULL**)

- ❖ **NULL** : n'est pas une valeur mais une *absence de valeur*! Les opérations ou les comparaisons ne peuvent lui être appliqué

- ❖ Toute **opération** (+,-,/,*) appliquée à NULL donne **NULL**

- ❖ Toute comparaison avec NULL donne ni vrai ni faux, mais INCONNU

Syntaxe du tri

```
SELECT col1, ...coln  
FROM nomtable  
WHERE condition  
ORDER BY coli, ..., colk
```

- Dans la clause ORDER BY, on peut avoir des :
 - des noms de colonnes : col₁, ...col_n
 - des expressions avec noms de colonnes
 - des numéros de position des colonnes dans la clause SELECT.
- On précise le sens : **ASC** (par défaut) ou **DESC**
- Les **valeurs nulles** sont **à la fin** par ordre **croissant**, au **début** par ordre **décroissant**.

Exemple de tri

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Noms, budgets et villes des projets de budget supérieur à 250 000, en ordonnant le résultat par ordre décroissant de budget puis par nom par ordre alphanumérique croissant ?

Exemple de tri

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Noms, budgets et villes des projets de budget supérieur à 250 000, en ordonnant le résultat par ordre décroissant de budget puis par nom par ordre alphanumérique croissant ?

```
SELECT Pname, Budget, City  
FROM Project  
WHERE Budget > 250000  
ORDER BY Budget DESC, Pname
```

Exemple de tri (2)

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Noms, budgets TTC (TVA 20%) et villes des projets, en ordonnant le résultat par ordre décroissant de budget TTC?

```
SELECT Pname, Budget*1.2, City
```

```
FROM Project
```

```
WHERE Budget > 250000
```

```
ORDER BY 2 DESC
```

- Noms, budgets et villes des projets en ordonnant le résultat par ordre décroissant de budget TTC?

```
SELECT Pname, Budget, City
```

```
FROM Project
```

```
WHERE Budget > 250000
```

```
ORDER BY Budget*1.2 DESC
```

Opérations ensemblistes

On peut réaliser des opérations ensemblistes sur les clauses SELECT.

3 opérations ensemblistes

UNION union de deux ensembles

INTERSECT intersection de deux ensembles

MINUS différence de deux ensembles (norme : **EXCEPT**)

Principe

Pour les opérations ensemblistes :

- Pas de lien entre les objets sélectionnés dans les 2 requêtes
- Même schéma dans les SELECT des deux requêtes : c'est à dire même nombre d'attributs et chacun du même type (par forcément le même nom)
- Le schéma en sortie correspond au schéma de la première requête
- Par défaut, les opérations ensemblistes éliminent les doublons (ensemble). Pour garder les doublons (multi-ensemble), il faut ajouter ALL après l'opérateur : **UNION ALL, EXCEPT ALL, INTERSECT ALL**

UNION

Emp(Eno, Ename, Title, City)
Pay(Title, Salary)

Project(Pno, Pname, Budget, Town)
Works(Eno, Pno, Resp, Dur)

- Noms des villes où habitent des employés ou où sont localisés des projets?

SELECT City

FROM Emp

UNION

SELECT Town

FROM Projet

INTERSECTION

Emp(Eno, Ename, Title, City)

Pay(Title, Salary)

Project(Pno, Pname, Budget, City)

Works(Eno, Pno, Resp, Dur)

- Noms des villes où habitent des employés et où sont localisés des projets?

SELECT City

FROM Emp

INTERSECT

SELECT Town

FROM Projet

DIFFERENCE

Emp(Eno, Ename, Title, City)
Pay(Title, Salary)

Project(Pno, Pname, Budget, City)
Works(Eno, Pno, Resp, Dur)

- Noms des villes où habitent des employés mais où n'est localisé aucun projet?

```
SELECT City
FROM Emp
MINUS
SELECT Town
FROM Projet
```