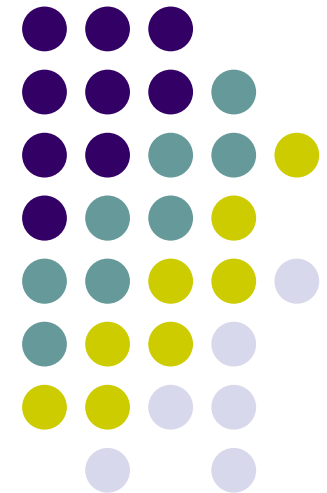


Le Langage d'Interrogation des Données Introduction

Plan :

- L'instruction SELECT
- Utilisation du Dictionnaire des Données





PRESENTATION

- Fonctions essentielles
 - Ecriture de requêtes de consultation de la base de données
 - Le LID ne permet pas de créer de nouvelles relations !!
- Principales instructions
 - Requêtes mono-tables :
 - **SELECT** Sélection, projection
 - Requêtes multi-tables
 - **SELECT** Jointures
 - **UNION, INTERSECT, MINUS** ensemblistes



SELECT : syntaxe

```
SELECT [DISTINCT/ALL] <colonne>  
FROM <table>  
[WHERE <condition>]  
[GROUP BY <colonne>|<expression>]  
[HAVING <condition>]  
[ORDER BY <colonne>|<expression> [asc|desc]] ;
```

- **[DISTINCT/ALL] :**
 - élimination ou conservation des tuples identiques dans le résultat (ALL par défaut)
- **<expression> :**
 - expression arithmétique, résultat de calcul ou de fonction
- **WHERE <condition> :**
 - sélection de tuples
- **[GROUP BY] :**
 - constitution de groupes de tuples ayant des valeurs identiques
- **[HAVING <condition>] :**
 - condition sur les groupes précédents
- **[ORDER BY] :**
 - affichage des tuples dans un ordre donné



Clause SELECT

- Dans la partie **SELECT** de la requête
 - Liste des colonnes désirées
 - * pour toutes les colonnes
 - Expression arithmétique, résultat d'une fonction

SELECT * FROM Employes;

IdEmploye	NomEmploye	Salaire	DateEmbauche	#IdService
18	Martin	30000	01/02/1999	2
12	Cartaux	25500	30/03/2000	3
23	Parot	23000	23/06/2001	3

SELECT NomEmploye FROM Employes;

NomEmploye
Martin
Cartaux
Parot



Clause SELECT

- Instruction **DISTINCT**
 - Permet d'éliminer les doublons, car SQL affiche le résultat d'une requête sans éliminer les répétitions

```
SELECT idService FROM Employes;
```

#IdService
2
3
3

```
SELECT DISTINCT idService FROM Employes;
```

#IdService
2
3



Clause WHERE

Conditions : forme générale

<opérande> <opérateur> <opérande>

- Opérandes et opérateurs doivent être compatibles
- On peut regrouper les conditions grâce aux opérateurs :
 - OR
 - AND
- Utilisation des parenthèses changer la priorité des opérateurs



Conditions : opérandes

- Il peut s'agir :
 - D'un nom de colonne
 - D'une valeur
 - Numérique (entière ou décimale)
 - Alphanumérique (entre ' ' dans Oracle)
 - Date (entre ' ' dans Oracle)
 - D'une expression
 - Arithmétique (+ - * /)
 - Utilisant des fonctions
 - D'une sous-requête
 - Requête fournissant une valeur ou une liste de valeurs compatibles



Conditions : opérateurs

- Opérateurs de comparaison classiques

= <> != > >= < <=

```
SELECT Nufact FROM Detail  
WHERE Copro='P03' AND Qte>20;
```

Copro
type varchar2

Montant
Type Number

- + beaucoup d'autres que vous verrez plus tard !



Exemples de requêtes SELECT

```
SELECT * FROM Employes WHERE salaire=30000;
```

IdEmploye	NomEmploye	Salaire	DateEmbauche	#IdService
18	Martin	30000	01/02/1999	2

```
SELECT IdEmploye, NomEmploye FROM Employes  
WHERE salaire=30000;
```

IdEmploye	NomEmploye
18	Martin



Exemples de requêtes SELECT

```
SELECT NomEmploye FROM Employes  
WHERE salaire > 25000  
and dateEmbauche > '01/01/2000';
```

AND

IdEmploye	NomEmploye	Salaire	DateEmbauche	#IdService
18	Martin	30000	01/02/1999	2
12	Cartaux	25500	30/03/2000	3
23	Parot	23000	23/06/2001	3



Requêtes multi-tables

- Dans la partie **SELECT** de la requête
 - Liste des colonnes désirées (nomtable.colonne si la colonne apparaît dans plusieurs tables de la requête)
 - * pour toutes les colonnes de toutes les tables de la requête
 - nomtable.* pour toutes les colonnes de la table nomtable
- Dans la partie **FROM** de la requête
 - Liste des tables pour lesquelles des colonnes sont utilisées dans la partie SELECT ou la partie WHERE
- Dans la partie **WHERE** de la requête
 - Conditions Table2.CléÉtrangère = Table1.CléPrimaire pour lier les tables du FROM combinées par des **AND**. **Aucune table ne doit rester non liée !**
 - Éventuellement conditions classiques (vues précédemment)



Exemples de requêtes

```
SELECT NomEmploye, NomService  
FROM Employes, Services  
WHERE Employes.IdService = Services.IdService  
and dateEmbauche > '01/01/2000';
```

NomEmploye	NomService
Cartaux	Informatique
Parot	Informatique

```
SELECT NomEmploye, NomService  
FROM Employes, Services  
WHERE Employes.IdService = Services.IdService  
and salaire > 25000  
And NomService = 'Informatique';
```



Utilisation multiple d'une table

- Dans la partie **FROM** de la requête
 - Ajout d'alias différents pour les différentes indications de la même table (nomtable1 alias1, nomtable alias2 ...)
- Dans la partie **SELECT** de la requête
 - Ajout des alias devant les colonnes (alias.colonne)
 - alias.* pour toutes les colonnes de la table associée à l'alias
 - * pour toutes les colonnes de toutes les tables de la requête
- Dans la partie **WHERE** de la requête
 - Ajout des alias devant les colonnes (alias.colonne)

Exemple :

```
SELECT Emp1.NomEmploye, Emp2.NomEmploye
FROM Employes Emp1, Employes Emp2
WHERE Emp1.IdService = Emp2.IdService
and Emp1.IdEmploye <> Emp2.IdEmploye ;
```



Rappel : Dictionnaire des Données Oracle

- Ensemble de tables et de vues utilisé en tant que référence en lecture seule.
- Stocke
 - les informations concernant les structures logique et physique de la base.
 - les utilisateurs valides d'une base Oracle
 - les informations concernant les contraintes d'intégrité définies sur les tables
 - l'espace alloué pour un objet du schéma ainsi que l'espace réellement utilisé
- Un dictionnaire de données est créé dès la création de la base
- Automatiquement mis à jour par Oracle en réponse à des actions spécifiques (telles que lorsque la structure de la base est altérée)



Vues du dictionnaire

- Une (méta) description du dictionnaire est donnée par la vue DICT.
Pour voir toutes les vues du dictionnaire :

```
SELECT * from DICT;
```

- Vues les plus utilisées pour l'utilisateur courant

NOM	ALIAS	Description
USER_TABLES	TABS	Mes tables : nom, statistiques, ..
USER_TAB_COLUMNS	COLS	Colonnes de mes tables : Nom colonne, type, longueur, ...
USER_CONSTRAINTS	-	Mes contraintes : Nom, type, table d'accueil, statut
USER_CONS_COLUMNS		Les colonnes sur lesquelles portent mes contraintes



Exemples d'utilisation

- Pour avoir des informations sur toutes les tables de l'utilisateur courant :

```
SELECT * from USER_TABLES;
```

- Pour avoir seulement le nom des tables de l'utilisateur courant :

```
SELECT table_name from USER_TABLES;
```

- Pour avoir toutes les colonnes de la table Employes :

```
SELECT * from USER_TAB_COLUMNS  
WHERE table_name='EMPLOYES';
```

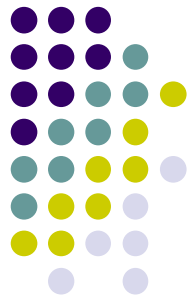



Autre commande très utile

- Pour voir la description d'une table ou d'une vue (que ce soit une table de l'utilisateur courant ou une vue du dictionnaire des données) :

```
DESC EMPLOYES ;
```

```
DESC USER_TABLES ;
```



Vues du dictionnaire

- Vues les plus utilisées pour obtenir des informations sur toutes les bases accessibles par l'utilisateur courant (y compris celles des autres utilisateurs)

NOM	Description
ALL_TABLES	Tables accessibles : nom, statistiques...
ALL_TAB_COLUMNS	Colonnes des tables : Nom colonne, type, longueur...
ALL_CONSTRAINTS	Contraintes : Nom, type, table d'accueil, statut
ALL_CONS_COLUMNS	Les colonnes sur lesquelles portent les contraintes

- Mêmes colonnes qu'avec le préfixe USER_ avec en plus une colonne OWNER



Exemples d'utilisation

- Pour avoir des informations sur toutes les tables accessibles par l'utilisateur courant :

```
SELECT * from ALL_TABLES;
```

- Pour avoir seulement le nom des tables de l'utilisateur kelkin :

```
SELECT table_name from ALL_TABLES  
WHERE OWNER='KELKIN' ;
```

Attention : le nom de l'utilisateur doit être en majuscules



Exemples d'utilisation

- Pour avoir toutes les colonnes de la table Employes de l'utilisateur kelkin :

```
SELECT * from ALL_TAB_COLUMNS  
WHERE table_name='EMPLOYES'  
AND OWNER='KELKIN' ;
```

Attention : les noms de l'utilisateur et de la table doivent être en majuscules

- Pour voir la description d'une table de l'utilisateur kelkin, accessible par l'utilisateur courant :

```
DESC kelkin.Employes ;
```