Chapitre 6:

SQL:Langage d'interrogation des données

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Objectifs du cours.

1. Définir le LID
2. Savoir formuler des Requêtes simples « SELECT »
3. Distinguer les fonctions mono lignes/multilignes
4. Définir les Jointures, les Opérateurs ensemblistes et les sous-interrogations

1ère Partie :

Les Requêtes simples « la clause SELECT »

Objectifs de la partie :

Définir «LID», savoir sélectionner des colonnes avec ou sans doublons, avec ou sans des conditions connaître les Expressions arithmétiques, la Valeur « NULL », les Alias de colonne, les opérateurs logique et ceux de comparaison.

LID.

Le langage LID permet, comme son nom l'indique, d'interroger une BD. Il sert à rechercher, extraire, trier, mettre en forme des données et calculer d'autres données à partir des données existantes.

La structure de base d'une interrogation est formée des 3 clauses suivantes :

SELECT *|<liste champ(s)> **FROM** <nom_table>
WHERE <condition(s)> ;

LID. (2/2)

- La clause SELECT : indique les colonnes à récupérer. (ou encore des champs projetés).
- Laclause FROM indique le nom de la ou des table(s) impliquée(s) dans l'interrogation.
- La clause WHERE correspond aux conditions de sélection des champs.
- Chaque nom de champ ou de table est séparé par une virgule.

Sélection des colonnes. Toutes les colonnes

```
SELECT *
FROM employees;
```

Affiche tous les champs de la table

Résultat:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
100	Steven	King	SKING	515.123.4567	17/06/87	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/89	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/93	AD_VP
103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/90	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	21/05/91	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	25/06/97	IT_PROG

Sélection des colonnes. Une/plusieurs colonnes

```
SELECT first_name, last_name
FROM employees;
```

Affiche les champs : « last_name » et « first_name »

Résultat:

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates

Expressions arithmétiques. Définition

i. Une expression contient des données de type NUMBER, DATE et des opérateurs arithmétiques.

Opérateur	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division

Expressions arithmétiques. Exemple 1

```
SELECT   min_salary + max_salary , min_salary - max_salary,
                 min_salary * max_salary, min_salary / max_salary
FROM   jobs;
```

Résultat:

MIN_SALARY+MAX_SALARY	MIN_SALARY-MAX_SALARY	MIN_SALARY*MAX_SALARY	MIN_SALARY/MAX_SALARY
60000	-20000	800000000	,5
45000	-15000	450000000	,5
9000	-3000	18000000	,5
24200	-7800	131200000	,5125
13200	-4800	37800000	,46666666666666666666666666666666666
24200	-7800	131200000	,5125
13200	-4800	37800000	,46666666666666666666666666666666666

Expressions arithmétiques. Exemple 2

Ecrire la requête qui permet d'afficher le nom, prénom de tous les employés ainsi que leur salaire annuel.

FIRST_NAME	LAST_NAME	Salaire Annuel
Steven	King	288600
Neena	Kochhar	204600
Lex	De Haan	204600
Alexander	Hunold	108600
Bruce	Ernst	72600
David	Austin	58200
Valli	Pataballa	58200
Diana	Lorentz	51000

Valeur «NULL».

Définition

- NULL est une valeur qui n'est pas disponible, non affectée ou inconnue.
- NULL est différent de zéro, espace ou chaîne vide

Exemple 1:

```
SELECT employee_id, commission_pct
FROM employees ;
```

Résultat:

EMPLOYEE_ID	COMMISSION_PCT
100	-
101	-
102	-
103	-
104	-
105	-

Valeur «NULL». Dans les expressions arithmétique.

Exemple 2 :

```
SELECT Employee_id, (1+commission_pct)*salary
FROM employees;
```

Résultat:

EMPLOYEE_ID	(1+COMMISSION_PCT)*SALARY
100	-
101	-
102	-
103	-
104	-
105	

Alias de colonne.

- Renomme un en-tête de colonne
- Suit le nom de colonne
- Peut utiliser le mot clé « as »
- Doit obligatoirement être inclus entre guillemets s'il contient des espaces, des caractères spéciaux ou s'il y a des majuscules et des minuscules

Alias de colonne. Exemple (2/2)

```
SELECT first_name as nom, last_name prenom
FROM employees;
```

Résultat:

NOM	PRENOM
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer

```
SELECT first_name as "nom de l'employee", last_name "prenom de l'employee"
FROM employees;
```

Résultat:

Nom De L'employee	Prénom De L'employee
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer

Opérateur de concaténation.

- Concatène des colonnes et/ou des chaines de caractères
- Est représenté par le symbole ||

Exemple:

```
SELECT 'le nom est ' ||first_name|| ', le prénom est '||last_name as "nom et
  prenom de l'employe"
FROM employees;
```

Résultat:

Nom Et Prénom De L'employé
le nom est Ellen, le prénom est Abel
le nom est Sundar, le prénom est Ande
le nom est Mozhe, le prénom est Atkinson
le nom est David, le prénom est Austin

Eliminer les doublons

- le mot-clé DISTINCT élimine les doublons.

Exemple:

```
SELECT DISTINCT(commission_pct)
FROM employees;
```

Résultat:

COMMISSION_PCT
-
,15
,35
,4
,3
,2
,25
,1

Restriction avec la clause

« WHERE »

Syntaxe :

SELECT<nom_des colonnes> FROM <nom_de_la_table >

WHERE <conditions>;

Exemple: liste des employés du département 20

```
SELECT last_name , first_name FROM employees
WHERE department_id=20;
```

Résultat:

LAST_NAME	FIRST_NAME
Hartstein	Michael
Fay	Pat

2 lignes renvoyées en 0,05 secondes

Opérateurs de comparaison.

(1/2)

OPERATEUR	DESCRIPTION
=	Egal à
<	Inférieur à
<=	Inférieur à ou égal
>	Supérieur à
>=	Supérieur à ou égale à
<> Ou !=	Différent
BETWEEN val1 AND val2	val1 <= val <= val2
In (liste de valeurs)	Valeur de la liste
LIKE (_ :un caractère, %=plusieurs caractères)	Comme
IS NULL	Correspond à une valeur NULL

Opérateurs logiques. (2/2)

OPERATEUR	DESCRIPTION
AND	Retourne TRUE si les deux conditions sont VRAIES
OR	Retourne TRUE si au moins une des conditions est VRAIE
NOT	Inverse la valeur de la condition TRUE si la condition est FAUSSE FALSE si la condition est VRAIE

Trier avec « ORDER by ».

Définition

Triez les lignes selon un ordre bien précis avec la clause ORDER BY:

- ASC : ordre croissant (par défaut)
- DESC : ordre décroissant

Remarque: La clause ORDER BY vient en dernier dans l'instruction SELECT

Trier avec « ORDER by ». Exemple

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC;
```

Résultat:

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Kumar	SA_REP	80	21/04/00
Banda	SA_REP	80	21/04/00
Ande	SA_REP	80	24/03/00
Markle	ST_CLERK	50	08/03/00
Lee	SA_REP	80	23/02/00
Philtanker	ST_CLERK	50	06/02/00

Exercices d’applications. (1/2)

(nous allons utilisé la table « employees » sous le schèma HR)

- Afficher la liste de tous les employés.
- Afficher le nom, le prénom, la date d’embauche de tous les employés et renommer les colonnes comme suit: « nom de l’employe », « prenom de l’employe » et « date dembauche »
- Afficher le nom et le prénom des employés triés par date d’embauche.
- Afficher les employés qui ont été embauché au cours de l’année 1990.
- Afficher la liste des employés dont le salaire est > 2800.
- Afficher la liste des employés des départements 10, 30 et 50.

Exercices d’applications. (2/2)

- Afficher la liste des employés dont le salaire entre 4000 et 6000 du département 20
  - Afficher la liste des employés dont la commission_pct est NULL.
  - Afficher la liste des employés dont le nom commence par la lettre ‘A’
  - Afficher la liste des employés dont le prénom comporte la lettre ‘s’
  - Afficher la liste des employés dont la deuxième lettre du nom est ‘i’
- Afficher la liste des employés embauchés pendant les années entre 1986 et 1990
- Afficher la liste des employés dont le job est différent de 'CLERK‘ ou 'MANAGER'