

La commande SELECT sur une table (suite)

Expression avec une colonne nullable
Tableau des fonctions
Recherche d'informations
Fonctions scalaires, fonctions statistiques
GROUP BY
HAVING
ORDER

XH

1

1

Expressions et conditions sur les valeurs nulles (NULL)

- Arithmétique
 - opérande NULL => UNKNOWN
- Comparaison (>, <, =, ...)
 - opérande NULL => UNKNOWN
- Table de vérité pour AND/OR/NOT

x	y	x AND y	x OR y	NOT x
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	UNKNOWN	UNKNOWN	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
UNKNOWN	TRUE	UNKNOWN	TRUE	UNKNOWN
UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN
UNKNOWN	FALSE	FALSE	UNKNOWN	UNKNOWN
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	UNKNOWN	FALSE	UNKNOWN	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

XH

2

2



Exemple

- Si x est NULL

```
SELECT *  
FROM T  
WHERE x = 0 OR x <> 0
```

- Est-ce que j'afficherais la ligne lorsque x vaut null ?
 - UNKNOWN OR UNKNOWN = UNKNOWN
 - pas dans le résultat !

XH

3

3



Quelques fonctions simples (scalaires) SQL2-ANSI

- POSITION(patron IN chaîne)
- CHARACTER_LENGTH(chaîne)
- OCTET_LENGTH (chaîne)
- BIT_LENGTH(chaîne)
- EXTRACT(champ FROM dateOuTime)
- SUBSTRING (chaîne FROM indiceDébut FOR nombreCaractères)
- UPPER | LOWER (chaîne)
- TRIM ([LEADING|TRAILING|BOTH] caractère FROM chaîne)
- CAST(expression AS type)
- ...
- Bien sûr, voir la documentation propre au SGBD ...

XH

4

4












Documentation pour les fonctions

- Voir le document papier « fonctions, expression, opérateur et conditions »
- Voir votre document « SQL pour Oracle » page 39 « les fonctions »
- site officiel Oracle 10gR2

XH

5

5

Fonctions	SGBDR	Paradox 9	Access 2000	Sybase Adaptive	SQL Server 7	Oracle 8
 (concaténation)		oui	non (&)	non (+)	non (+)	oui
 CAST		oui	non	non (CONVERT)	oui	non (TO_CHAR / TO_NUM / TO_DATE)
 LIKE / ESCAPE / % / _		oui / oui / oui / oui	oui / non / non / non	oui / oui / oui / oui	oui / oui / oui / oui	oui / oui / oui / oui
 TRIM		oui	non (LTRIM, RTRIM)	non (LTRIM, RTRIM)	non (LTRIM, RTRIM)	non (LTRIM, RTRIM)
 SUBSTRING		oui	non (MID)	non (ODBC SUBSTRING)	non (ODBC SUBSTRING)	non (SUBSTR)
 LOWER / UPPER		oui / oui	non (UCASE) / non (LCASE)	oui / oui	oui / oui	oui / oui
 EXTRACT		oui	non	non (DATEPART)	non (DATEPART)	non
 CURRENT_DATE / CURRENT_TIME / CURRENT_TIMESTAMP		non / non / non (TODAY, NOW en QBE)	non (NOW)	non (GETDATE)	non (GETDATE()) / CURRENT_TIMESTAMP	non (SYSDATE)
 AVG / MAX / MIN / SUM / COUNT		oui / oui / oui / oui / oui	oui / oui / oui / oui / oui	oui / oui / oui / oui / oui	oui / oui / oui / oui / oui	oui / oui / oui / oui / oui
VARIANCE / STDEV		non / non	non / non	non / non	non / non	oui / oui

6

6

Fonctions	SGBDR	Paradox 9	Access 2000	Sybase Adaptive	SQL Server 7	Oracle 8
ABS / MOD / SIGN / SQRT / PI	non / non / non / non / non	non / non / non / non / non	oui / non / non (SGN) / non (SQR) / non	oui / non (%) / oui / oui / oui	oui / non (%) / oui / oui / oui	oui / oui / oui / oui / non
CEIL / FLOOR / ROUND / TRUNC	non / non / non / non	non / non (INT) / non / non	non / oui / oui / non	non / oui / oui / non	oui / oui / oui / oui	oui / oui / oui / oui
EXP / LN / LOG / POWER	non / non / non / non	oui / non / oui / non	oui / non / oui / oui	oui / non / oui / oui	oui / non / oui / oui	oui / oui / non (LOG (M, n)) / oui
COS / SIN / TAN	non / non / non	oui / oui / oui	oui / oui / oui	oui / oui / oui	oui / oui / oui	oui / oui / oui
COSH / SINH / TANH	non / non / non / non	non / non / non	non / non / non	non / non / non	non / non / non	oui / oui / oui (2)
INITCAP / LPAD / RPAD / REPLACE	non / non / non / non	non / non / non / non	non / non / non / non	non / non / non / non	non / non / non / oui	oui / oui / oui / oui
SOUNDEX / TRANSLATE / INSTR / LENGTH	non (COMME dans QBE) / non / non / non	non / non / non (LOCATE) / non	oui / non (REPLACE) / non (PATINDEX) / non (LEN)	oui / non (REPLACE) / non (PATINDEX) / non (LEN)	oui / oui / non (LOCATE) / non	oui / oui / non (LOCATE) / non
TO_CHAR pour numérique / pour date	non / non	non / non	non (CONVERT) / non (CONVERT)	non (CONVERT) / non (CONVERT)	non (auto (3)) / non	non
ASCII / CHR	non / non	non / non	oui / non (CHAR)	oui / non (CHAR)	oui / oui	oui / oui
ADD_MONTH / MONTH_BETWEEN	non / non	non (DATEADD) / non (DATEDIFF)	non (DATEADD) / non (DATEDIFF)	non (DATEADD) / non (DATEDIFF)	oui / oui	oui / oui
LAST_DAY / NEXT_DAY	non / non	non / non	non / non	non / non	oui / oui	oui / oui

7

Tableau des fonctions (3)

Fonctions	SGBDR	Paradox 9	Access 2000	Sybase Adaptive	SQL Server 7	Oracle 8
GREATEST / LEAST / DECODE	non / non / non	non / non / non	non / non / non	non / non / non	non / non / non	oui / oui / non
SQL 2 IS NULL / COALESCE / NULLIF	oui / non / non	oui / non / non (ISNULL)	oui / oui / non (ISNULL)	oui / oui / oui	oui / non / non (NVL)	oui
SQL 2 ROW VALUE CONSTRUCTOR	non	non	non	non	oui	oui

8



Les fonctions pour le temps dans Oracle

- Sur le site ftp, lisez le pdf de SOUTOU pour voir les différents types de champs disponibles

XH

9

9



Exemple d'expressions avec une DATE Oracle Fonction de conversion

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE  
-----  
02-02-05
```

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY HH24:MI:SS';
```

```
Session altered.
```

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE  
-----  
05-02-2002 09:08:26
```

```
SQL> SELECT TO_DATE('05/02/2000', 'DD/MM/YYYY') FROM DUAL;
```

```
TO_DATE('05/02/2000  
-----  
05-02-2000 00:00:00
```

```
SQL> SELECT TO_CHAR(SYSDATE, 'DD/MM/YYYY') FROM DUAL;
```

```
TO_CHAR(SY  
-----  
22/01/2002
```

XH

10

10



Opérations sur le type DATE

```
SQL> SELECT SYSDATE + INTERVAL '1' DAY FROM DUAL;

SYSDATE+INTERVAL'1'
-----
23-01-2002 16:02:18

SQL> SELECT SYSDATE - INTERVAL '1' DAY FROM DUAL;

SYSDATE-INTERVAL'1'
-----
21-01-2002 16:02:18

SQL> SELECT SYSDATE + 1 FROM DUAL;

SYSDATE+1
-----
23-01-2002 16:02:18

SQL> SELECT SYSDATE + 1/24 FROM DUAL;

SYSDATE+1/24
-----
22-01-2002 17:02:18

SQL> SELECT SYSDATE + INTERVAL '30' SECOND FROM DUAL;

SYSDATE+INTERVAL'30
-----
22-01-2002 16:02:48
```

XH

11

11



XH

12

12



Tri du résultat clause ORDER BY

- Les Clients en ordre alphanumérique du nom

```
SELECT      *  
FROM        Client  
ORDER BY    nomClient
```

- Tri multicritère

```
SELECT      *  
FROM        Client  
ORDER BY    nomClient DESC, noTéléphone ASC
```

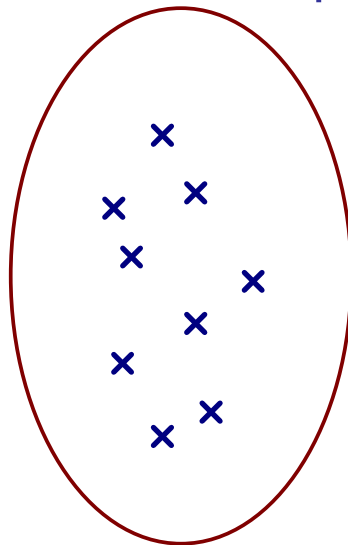
XH

13

13



Vision d'une table pour SQL




- On travaille sur des ensembles de tuples

XH

14


14



Les Fonctions statistiques

14/12/2022 XH 15

15



Fonctions statistiques

- Count
- Avg
- Sum
- Min
- max

XH 16

16



Exemple

```
select count(refart), max(prix), avg(qtestk)
from articles;
```

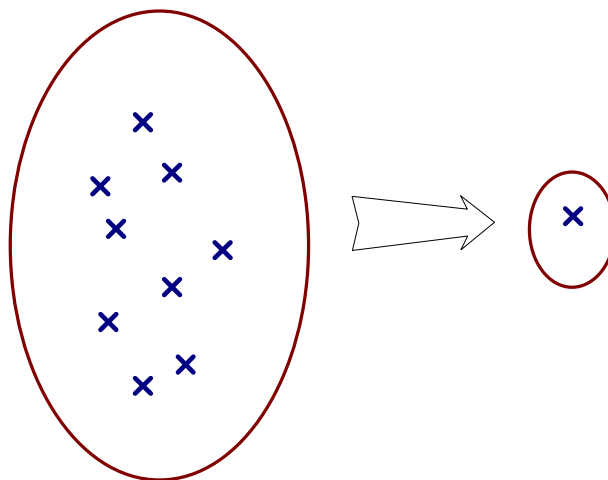
XH

17

17



Ce qu'il faut voir
pour une fonction statistique (sans
group by)



XH

18

18



Autre exemple avec fonction statistique

- Le nombre d'*Articles* différents à vendre ainsi que le *prixUnitaire* moyen des *Articles*

```
SELECT COUNT(*) AS nombreArticles,  
       AVG(prixUnitaire) AS prixMoyen  
FROM Article
```

nombreArticles	prixMoyen
10	19.49

Autres noms: Fonctions de groupe, Fonctions d'agrégations

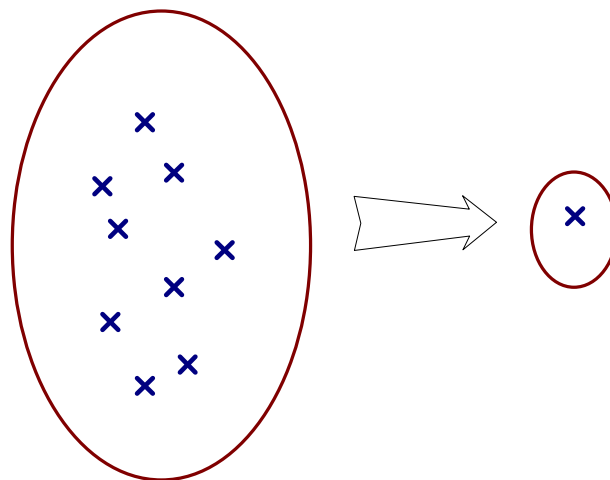
XH

19

19



Ce qu'il faut voir
pour une fonction statistique (sans
group by)



XH

20

20



Exemple avec fonction statistique(2)

```
SELECT Count(DISTINCT prixUnitaire) AS nombrePrix  
FROM Article
```

nombrePrix
6

```
SELECT Count(prixUnitaire) AS nombrePrixNonNull  
FROM Article
```

nombrePrixNonNull
10

XH

21

21



Les clauses group by, having et order by

14/12/2022

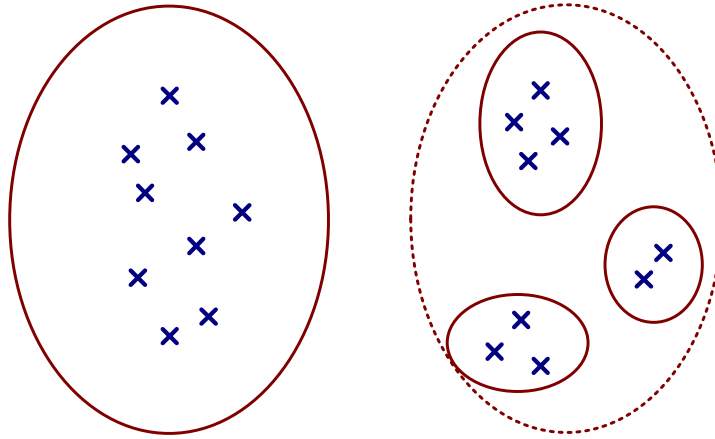
XH

22

22



Ce qu'il faut voir pour le regroupement



XH

23

23



clause GROUP BY Partition d'une table avec la

- Produire le nombre de *Commandes* passées par chacun des *Clients* qui ont passé au moins une *Commande*

```
SELECT    noClient, COUNT(*) AS nombreCommandes
FROM      Commande
GROUP BY  noClient
```

Table <i>Commande</i>		
noCommande	dateCommande	noClient
1	01/06/2000	10
3	02/06/2000	10
4	05/07/2000	10
2	02/06/2000	20
6	09/07/2000	20
5	09/07/2000	30
7	15/07/2000	40
8	15/07/2000	40

noClient	nombreCommandes
10	3
20	2
30	1
40	2

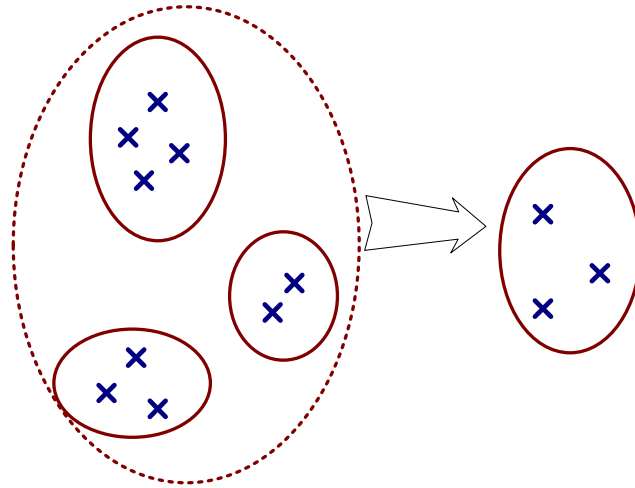
XH

24

24



Ce qu'il faut voir
pour la fonction statistique appliquée à
chaque groupe



XH

25

25



clause GROUP BY (suite)

- Pour chacune des LigneCommande pour lesquelles au moins une Livraison a été effectuée, afficher le noCommande et noArticle, la quantité totale livrée et le nombre de Livraisons effectuées

```
SELECT      noCommande, noArticle, SUM(quantitéLivrée) AS totalLivré,  
            COUNT(*) AS nombreLivraisons  
FROM        DétailLivraison  
GROUP BY    noCommande, noArticle
```

Table DétailLivraison			
noLivraison	noCommande	noArticle	quantitéLivrée
100	1	10	7
101	1	10	3
100	1	70	5
102	2	40	2
102	2	95	1
100	3	20	1
103	1	90	1
104	4	40	1
105	5	70	2

XH

noCommande	noArticle	totalLivré	nombreLivraisons
1	10	10	2
1	70	5	1
1	90	1	1
2	40	2	1
2	95	1	1
3	20	1	1
4	40	1	1
5	70	2	1

26

26



Clause HAVING

- Produire le nombre de *Commandes* passées par chacun des *Clients* qui ont passé deux *Commandes* ou plus

```
SELECT  noClient, COUNT(*) AS nombreCommandes
FROM    Commande
GROUP BY noClient
HAVING  COUNT(*) >= 2
```

noCommande	dateCommande	noClient
1	01/06/2000	10
3	02/06/2000	10
4	05/07/2000	10
2	02/06/2000	20
6	09/07/2000	20
5	09/07/2000	30
7	15/07/2000	40
8	15/07/2000	40

noClient	nombreCommandes
10	3
20	2
30	1
40	2

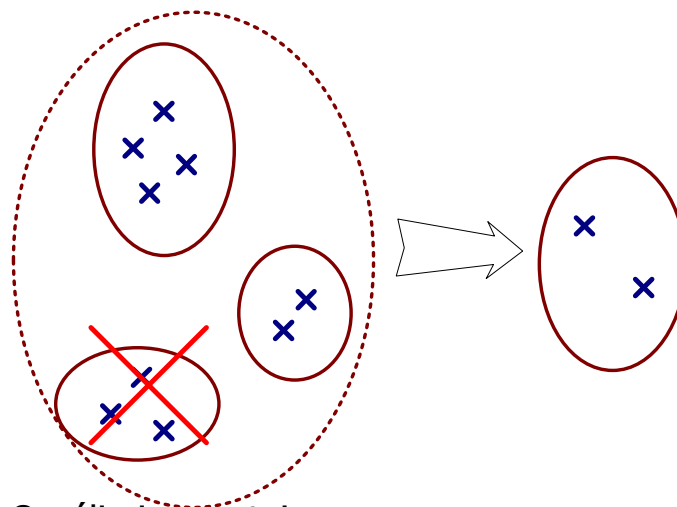
XH

27

27



Ce qu'il faut voir avec le having



- On élimine certains groupes

XH

28

28



Clause HAVING (suite)

Produire le nombre de *Commandes* passées par chacun des *Clients* qui ont passé deux *Commandes* ou plus après le 02/06/2000

```
SELECT      noClient, COUNT(*) AS nombreCommandes
FROM        Commande
WHERE       dateCommande > '02/06/2000'
GROUP BY    noClient
HAVING      COUNT(*) >= 2
```

Table <i>Commande</i>		
noCommande	dateCommande	noClient
1	01/06/2000	10
3	02/06/2000	10
4	05/07/2000	10
2	02/06/2000	20
6	09/07/2000	20
5	09/07/2000	30
7	15/07/2000	40
8	15/07/2000	40

noClient	nombreCommandes
10	1
20	1
30	1
40	2

XH

29

29



END



XH

30

30