

Les fonctions monolignes

Fonctions de caractères

Fonctions de conversion majuscules/minuscules

Fonction	Exemple	Résultat	Description
LOWER(chaine)	LOWER('SQL Course')	sql course	convertit les caractères majuscules de «chaine» en minuscules
UPPER(chaine)	UPPER('SQL Course')	SQLCOURSE	convertit les caractères minuscules de «chaine» en majuscules
INITCAP(chaine)	INITCAP('SQL Course')	SqI Course	convertit l'initiale de chaque mot de «chaine» en majuscule et les caractères suivants en minuscules

Exemple

```
SELECT first_name, lower(first_name) , upper(first_name), initcap(first_name) FROM employees;
```

FIRST_NAME	LOWER(FIRST_NAME)	UPPER(FIRST_NAME)	INITCAP(FIRST_NAME)
Ellen	ellen	ELLEN	Ellen
Sundar	sundar	SUNDAR	Sundar
Mozhe	mozhe	MOZHE	Mozhe
David	david	DAVID	David
Hermann	hermann	HERMANN	Hermann
Shelli	shelli	SHELLI	Shelli

Fonctions de manipulation de caractères

Fonction	Exemple	Résultat	Description
CONCAT(chain1,chain2)	CONCAT('Hello', 'World')	HelloWorld	concatène la première «chain1» avec «chain2» (comme l'opérateur)
SUBSTR(chain, pos [, long])	SUBSTR('HelloWorld', 1, 5)	Hello	extrait une sous chaîne de «chain» à partir de la position «pos» et de longueur «long»
LENGTH(chain)	LENGTH('HelloWorld')	10	taille d'une chaîne de caractères
INSTR(chain1,chain2)	INSTR('HelloWorld', 'W')	6	position de «chain2» dans «chain1»
LPAD(chain, long [, car])	LPAD(salary, 10, '')	*****24000	complète (ou tronque) «chain» à gauche à la longueur «long» par le caractère «car» (espace par défaut)
RPAD(chain, long [, car])	RPAD(salary, 10, '')	24000****	complète (ou tronque) «chain» à droite à la longueur «long» par le caractère «car» (espace par défaut)
LTRIM(chain [, car])	LTRIM('abbabaa ','a')	bbabaa	supprime les caractères à gauche de la chaîne «chain» tant qu'ils appartiennent à l'ensemble de caractères «car» (espace par défaut)
RTRIM(chain [, car])	RTRIM('abbabaa ','a')	abbab	supprime les caractères à droite de la chaîne «chain» tant qu'ils appartiennent à l'ensemble de caractères «car» (espace par défaut)
REPLACE(chain, ch1, ch2)	REPLACE('JACK and JUE','J','BL')	BLACK and BLUE	remplace toutes les occurrences de «ch1» dans «chain» par «ch2»
TRANSLATE(chain, ch1, ch2)	TRANSLATE('JACK and JUE','J','BL')	BACK and BUE	remplace chaque caractère de «ch1» dans «chain» par son correspondant dans «ch2»
ASCII(chain)	ASCII('Black')	66	retourne le code ASCII du premier caractère de «chain»
CHR(n)	CHR(98)	b	retourne le caractère ayant le code ASCII «n»

Exemple

```

SELECT first_name, last_name, job_id,
CONCAT(first_name,last_name) "Nom et prénom",
LENGTH (last_name) "longueur nom",
INSTR(last_name,'a') "position a",
LPAD(last_name,10,'*'),
RPAD(last_name,10,'*')
FROM employees
WHERE SUBSTR(job_id, 4) = 'REP';

```

FIRST_NAME	LAST_NAME	JOB_ID	Nom Et Prénom	Longueur Nom	Position A	LPAD(LAST_NAME,10,'*')	RPAD(LAST_NAME,10,'*')
Peter	Tucker	SA REP	PeterTucker	6	0	****Tucker	Tucker***
David	Bernstein	SA REP	DavidBernstein	9	0	*Bernstein	Bernstein*
Peter	Hall	SA REP	PeterHall	4	2	****Hall	Hall****
Christopher	Olsen	SA REP	ChristopherOlsen	5	0	****Olsen	Olsen****
Nanette	Cambrault	SA REP	NanetteCambrault	9	2	*Cambrault	Cambrault*
Oliver	Tuvault	SA REP	OliverTuvault	7	4	**Tuvault	Tuvault**
Janette	King	SA REP	JanetteKing	4	0	****King	King*****
Patrick	Sully	SA REP	PatrickSully	5	0	****Sully	Sully****
Allan	McEwen	SA REP	AllanMcEwen	6	0	***McEwen	McEwen***

Fonctions numériques

Fonction	Exemple	Résultat	Description
ABS(n)	ABS(-5)	5	Valeur absolue de n
MOD(n1, n2)	MOD(15,4)	3	reste de division de n1 par n2
POWER(n, e)	POWER(2,3)	8	n à la puissance e
SIGN(n)	SIGN(6)	1	Retourne -1 si n<0, 0 si n=0, 1 si n>0
SQRT(n)	SQRT(4)	16	racine carrée de n
ROUND(n[, p])	ROUND(45.926, 2) ROUND(45.924, 2) ROUND(-45.925,2)	45.93 45.92 -45.93	arrondit n à la précision p (0 par défaut)
TRUNC(n[, p])	TRUNC(45.926, 2) TRUNC(45.924, 2) TRUNC(-45.925,2)	45.92 45.92 -45.92	tronque n à la précision p (0 par défaut)
FLOOR(n)	FLOOR(45.926) FLOOR(-45.926)	45 -46	retourne la partie entière inférieure (si n<x<n+1 alors FLOOR(x)=n)
CEIL(n)	CEIL(45.926) CEIL(-45.926)	46 -45	retourne la partie entière supérieure (si n<x<n+1 alors CEIL(x)=n+1)
GREATEST(n1, n2,...)	GREATEST(4,-1,7)	7	maximum de n1, n2,...
LEAST(n1, n2,...)	LEAST(4,-1,7)	-1	minimum de n1, n2,...

Exemple

```
SELECT commission_pct+0.2,
ROUND(commission_pct+0.2),
TRUNC(commission_pct+0.2),
FLOOR(commission_pct+0.2),
CEIL(commission_pct+0.2)
FROM employees where department_id=80;
```

COMMISSION_PCT+0.2	ROUNDD(COMMISSION_PCT+0.2)	TRUNC(COMMISSION_PCT+0.2)	FLOOR(COMMISSION_PCT+0.2)	CEIL(COMMISSION_PCT+0.2)
,6	1	0	0	1
,5	1	0	0	1
,5	1	0	0	1
,5	1	0	0	1
,4	0	0	0	1
,5	1	0	0	1
,45	0	0	0	1

Fonctions de dates

- Dans la base de données Oracle, les dates sont stockées dans un format numériques interne: siècle, année, mois, jour, heures, minutes et secondes
- Le format de date par défaut est '**DD-MON-YY**'
- SYSDATE** est une fonction qui renvoie :
 - La date**
 - L'heure**

Calcul arithmétique sur les dates

Calcul arithmétique sur des dates:

- ajout ou soustraction d'un nombre de jour à une date an d'obtenir une date résultante
 - Ajout ou soustraction d'un nombre d'heures à une date en divisant le nombre d'heures par 24
 - soustraction d'une date d'une autre an de déterminer le nombre de jours entre les deux dates.

Exemple

```
SELECT first_name,  
(SYSDATE-hire_date) AS jours,  
(SYSDATE-hire_date)/7 AS semaines  
FROM employees;
```

Fonction	Exemple	Résultat	Description
SYSDATE ou CURRENT_DATE			retourne la date et l'heure courante du système d'exploitation hôte
MONTHS_BETWEEN(d1,d2)	MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')	19.6774194	Nombre de mois entre deux dates d1 et d2
ADD_MONTHS(d,j)	ADD_MONTHS ('11-JAN-94',6)	11-JUL-94	ajoute j mois à une date d
NEXT_DAY(d,j)	NEXT_DAY('01-SEP-95','FRIDAY')	08-SEP-95	retourne la date du jour j qui suit la date d
LAST_DAY(d)	LAST_DAY('01-FEB-95')	28-FEB-95	retourne le dernier jour du mois de la date d
ROUND(d, p)	<ul style="list-style-type: none"> •ROUND(to_date('25-07-03','DD-MM-YY'),'MONTH') •ROUND(to_date('25-07-03','DD-MM-YY'),'YEAR') 	01-08-03 01-01-04	arrondit la date d à la précision p. La précision est indiquée en utilisant un des masques de mise en forme de la date
TRUNC(d, p)	<ul style="list-style-type: none"> •TRUNC(to_date('25-07-03','DD-MM-YY'),'MONTH') •TRUNC(to_date('25-07-03','DD-MM-YY'),'YEAR') 	01-07-03 01-01-03	tronque la date d à la précision p
EXTRACT(p FROM d)	extract(day from to_date('25-07-03','DD-MM-YY'))	25	extrait la précision p de la date d

Exemple

```
SELECT systimestamp,  
extract(year from systimestamp) as Année,  
extract(month from systimestamp) as Mois,  
extract(day from systimestamp) as Jour,  
extract(hour from systimestamp) as Hour,  
extract(minute from systimestamp) as Minutes,  
extract(second from systimestamp) as Secondes  
FROM dual;
```

SYSTIMESTAMP	ANNÉE	MOIS	JOUR	HOUR	MINUTES	SECONDES
01-NOV. -13 12.48.03,354000 PM +01:00	2013	11	1	11	48	3,354

Fonctions de conversion

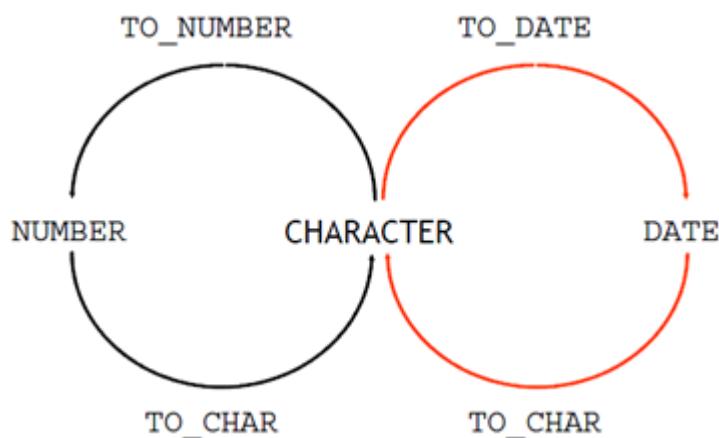
Conversion implicite

- Pour les affectations, le serveur Oracle peut convertir automatiquement les types de données suivants:

FROM	TO
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE

- L'expression salary='2000' entraîne la conversion implicite de la chaîne '2000' en valeur numérique 2000
- L'expression hire_date>'01-Jan-90' entraîne la conversion implicite de la chaîne '01-Jan-90' en date 37/90

Conversion explicite



Conversion explicite: fonction TO_CHAR

Fonction	Description
TO_CHAR(date, format)	<p>permet de convertir une date en chaîne de caractère en fonction d'un format</p> <ul style="list-style-type: none"> • YYYY année • YY deux derniers chiffres de l'année • WW numéro de la semaine dans l'année • MM numéro du mois • MONTH nom du mois • MON nom abrévié du mois • DDD numéro du jour dans l'année • DD numéro du jour dans le mois • D numéro du jour dans la semaine (1 pour lundi, 7 pour dimanche) • HH ou HH12 heure (sur 12 heures) • HH24 heure (sur 24 heures) • MI minutes • SS secondes
TO_CHAR(nombre, format)	<p>permet de convertir un nombre en chaîne de caractère en fonction d'un format</p> <ul style="list-style-type: none"> • 9 représente un chiffre (non représenté si non significatif) • 0 représente un chiffre (présent même si non significatif) • . point décimal apparent • , une virgule apparaîtra à cet endroit • \$ un \$ précédera le premier chiffre significatif • B le nombre sera représenté par des blancs s'il vaut zéro • MI le signe négatif sera à droite • PR un nombre négatif sera entre <>

Exemple 1

```
SELECT last_name, TO_CHAR(hire_date, 'fm DD Month YYYY') AS HIREDATE FROM employees;
```

LAST_NAME	HIREDATE
King	17 Juin 1987
Kochhar	21 Septembre 1989
De Haan	13 Janvier 1993
Hunold	3 Janvier 1990
Ernst	21 Mai 1991

- fm permet de supprimer les espaces de remplissage ou les zéros de début

Exemple 2

```
SELECT first_name, TO_CHAR(salary, '$99,000.00') SALARY FROM employees;
```

FIRST_NAME	SALARY
Steven	\$24,000.00
Nicola	\$17,000.00
Lay	\$17,000.00
Alexander	\$9,000.00
Bruce	\$8,000.00
David	\$4,800.00

Conversion explicite: fonctions TO_DATE / TO_NUMBER

Fonction	Description
TO_DATE (chaîne, format)	permet de convertir une chaîne de caractères en donnée de type date. Le format est identique à celui de la fonction TO_CHAR
TO_NUMBER (chaine, format)	convertit une chaîne de caractères en nombre dans le format donné (quand la chaîne de caractères est composée de caractères numériques)

Exemple 1

```
select first_name, hire_date from emp
where hire_date>to_date('01/01/1982','DD-MM-YYYY');
```

FIRST_NAME	HIRE_DATE
Michael	17/02/96
Pat	17/08/97

Exemple 2

```
Select first_name, salary from employees
where salary>=to_number('15000');
```

FIRST_NAME	SALARY
Steven	24000
Neena	17000
Lex	17000

Autres fonctions

Fonction DECODE

La fonction decode permet de faire un traitement conditionnel sur les données : Decode (expr, val1, res1, val2, res2, . . . ValN, resN, default) retourne res1 si expr = val1, res2 si expr=val2,...,resN si expr=valN sinon default.

Exemple

```
SELECT first_name,department_id, decode(department_id,10, 'ACCOUNTING', 20, 'RESEARCH', 'DEP. INCONNU ') AS "NOM DEPARTEMENT" FROM employees;
```

FIRST_NAME	DEPARTMENT_ID	NOM DEPARTEMENT
Jennifer	10	ACCOUNTING
Michael	20	RESEARCH
Pat	20	RESEARCH
Den	30	DEP. INCONNU
Alexander	30	DEP. INCONNU

Fonction NVL / NVL2

- NVL(expr,val): retourne val si expr est NULL
- NVL2(expr,val1,val2): retourne val1 si expr est NOT NULL et val2 si expr est NULL
- expr peut être de type date, les caractères et valeur numérique. Les types de données de expr et val doivent correspondre.

Exemple

```
SELECT first_name, salary, commission_pct,
NVL(commission_pct,0),
NVL(to_char(commission_pct), 'Pas de commission') AS "commission ?",
NVL2(commission_pct,commission_pct*salary,0) AS "commision",
to_char(NVL2(commission_pct,commission_pct*100/salary,0)) || '%' AS "pourcentage commission"
FROM employees;
```

Fonction NULLIF

NULLIF (expr1, expr2) : retourne NULL si expr1=expr2, sinon retourne expr1.

Exemple

```
SELECT first_name, LENGTH(first_name) nbr1, last_name, LENGTH(last_name) nbr2,  
NULLIF(LENGTH(first_name), LENGTH(last_name)) result FROM employees;
```

FIRST_NAME	NBR1	LAST_NAME	NBR2	RESULT
Amit	4	Banda	5	4
Sarah	5	Bell	4	5
Britney	7	Everett	7	-
Daniel	6	Faviet	6	-

Fonction COALESCE

Coalesce (expr1,expr2,expr3,...) : retourne la première valeur non nulle

Exemple

```
select Coalesce(NULL,1,NULL,7) from dual;  
=> retourne 1
```

Fonction CASE

La fonction `case` évalue une liste de conditions et retourne un résultat parmi les cas possibles.

- Case <expression>

when <valeur1> then <resultat1>

•

when <valeurN> then <resultatN>

else resultat

end

- Case

when <condition1> then <resultat1>

...

when <conditionN> then <resultatN>

else resultat

end

Exemple

```
SELECT first_name, department_id,  
case department_id  
when 10 then 'Accounting'  
when 20 then 'RESEARCH'  
else 'INCONNU'  
end as departement  
FROM employees;
```

FIRST_NAME	DEPARTMENT_ID	DEPARTEMENT
Jennifer	10	Accounting
Michael	20	RESEARCH
Pat	20	RESEARCH
Den	30	INCONNU
Alexander	30	INCONNU