## FONCTIONS DE CARACTÈRES:

|  |  |  |
| --- | --- | --- |
| FONCTION  ch = chaîne de caractères (mettre en guillemets)  n= nombre | valeur retournée | Exemple |
| CHR(n) | caractère équivalent à la valeur ASCII n | CHR(65) = A |
| CONCAT(ch1,ch2) | concaténation de ch1 avec ch2 |  |
| INITCAP(ch) | la première lettre de ch est capitalisée | INITCAP('livre') = Livre |
| LOWER (ch) | transforme tous les caractères en minuscule | LOWER('Livre') = livre |
| LPAD(ch1,n[,ch2])  ( L = left pour gauche) | ch1 est précédé de ch2 sur une longueur totale de n positions | LPAD('UN',7,'-') = -----UN |
| LTRIM (Ch,[,chaîne])  ( L = left pour gauche) | la première lettre de ch est enlevée si elle existe dans la chaîne. Le procédé continue jusqu'à ce qu'on rencontre une lettre n'existant pas dans la chaîne. | LTRIM('ECOLE','EC') = OLE |
| REPLACE( ch, |1,|2) | remplace toute occurrence |1 dans ch par |2. | REPLACE('SUCCEDE','SUC' 'PRE') = PRECEDE |
| RPAD(ch1,n[,ch2])  R = right pour droite | ch1 est suivi de ch2 sur une longueur totale de n positions | RPAD('UN',7,'-') = UN----- |
| RTRIM (Ch,[,chaîne])  ( R = right pour droite) | la dernière lettre de ch est enlevée si elle existe dans la chaîne. Le procédé continue jusqu'à ce qu'on rencontre une lettre n'existant pas dans la chaîne. | RTRIM('ECOLE','EC') = ECOL |
| SUBSTR(ch,m[,n]) | affiche n caractères de ch, commencant par le m ième caractère | SUBSTR('TRAVAIL',2,4)= RAVA |
| TRANSLATE(ch,ancienne,nouvelle) | remplace toute occurence ancienne de ch par nouvelle | SELECT poste, TRANSLATE(to\_char(sal),'5000','xxxx') salaire from emp  WHERE poste = 'president'; |
| UPPER(ch) | tous les caractères de ch sont en majuscule. | UPPER('exemple')= EXEMPLE |

## FONCTIONS DE CARACTÈRES RETOURNANT DES NOMBRES

EXEMPLES:

|  |  |  |
| --- | --- | --- |
| FONCTION  ch = chaîne de caractères | valeur retournée | Exemple |
| ASCII(ch) | valeur décimale du caractère | ASCII('Q') = 81 |
| LENGTH(ch) | longueur de ch en caractères | LENGTH('ECOLE') = 5 |
| INSTR(ch1,ch2¸[,N[,M]]) | Position de la m ième occurrence de ch2 dans ch1, commencant par le niène caractère. | INSTR('ECOLE','CO',1,1) =2  INSTR('COCO','CO',1,2) =3  INSTR('COICOI','OI',2,2)=5 |

## FONCTIONS DE DATE

Ces fonctions de date retournent des valeurs qui sont également de type DATE

|  |  |  |
| --- | --- | --- |
| Fonction | valeur retournée |  |
| ADD\_MONTHS  (date, n\_mois) | Ajoute n mois à la date et retourne une date comme résultat | SELECT ADD\_MONTHS (datembauche,12) from emp = |
| LAST\_DAY(date) | Dernier jour du mois qui contient la date | SELECT LAST\_DAY(SYSDATE) FROM DUAL= 03-01-31 |
| NEXT\_DAY(date,*char*) | Calcule la date du premier jour de la semaine indiqué par *char* qui suit la date. *Char* doit correspondre à un jour de la semaine | SELECT NEXT\_DAY('03-01-16','LUNDI') FROM DUAL= 03-01-20 |
| SYSDATE | date courante |  |
| ROUND(date,[,précision]) | La date est arrondie jusqu'à la précision spécifiée | SELECT ROUND(TO\_DATE('03-01-28'),'YEAR') FROM DUAL  = 03-01-01 |
| TRUNC(date,[,*format*]) | La date est tronquée à la précision spécifiée par *format* | SELECT TRUNC(TO\_DATE('03-02-10'),'DAY') FROM DUAL = 03-02-09 |

Tableau des précisions utilisées avec **ROUND et TRUNC**

|  |  |  |
| --- | --- | --- |
| Précision | Description | Exemple |
| CC,SCC | Siècle |  |
| YYYY,SYYYY  YEAR, SYEAR,  YYY, YY,Y | Année |  |
| Q | Trimestre (Quart de l'année: de 1 à 4)  Janvier à mars = 1 |  |
| MONTH, MON, MM | Mois (arrondi à partir du 16 ième jour) |  |
| DDD, DD, J | Jour |  |
| DAY, DY, D | Date du premier jour de la même semaine (un dimanche) | TRUNC(SYSDATE,'DAY')=  ROUND(SYSDATE,'DAY')= |

## FONCTIONS DE CARACTÈRES:

|  |  |  |
| --- | --- | --- |
| FONCTION  ch = chaîne de caractères (mettre en guillemets)  n= nombre | valeur retournée | Exemple |
| CHR(n) | caractère équivalent à la valeur ASCII n | CHR(75)= |
| CONCAT(ch1,ch2) | concaténation de ch1 avec ch2 |  |
| INITCAP(ch) | la première lettre de ch est capitalisée | INITCAP('livre') = |
| LOWER (ch) | transforme tous les caractères en minuscule | LOWER('Livre') = |
| LPAD(ch1,n[,ch2])  ( L = left pour gauche) | ch1 est précédé de ch2 sur une longueur totale de n positions | LPAD('UN',7,'-')= |
| LTRIM (Ch,[,chaîne])  ( L = left pour gauche) | la première lettre de ch est enlevée si elle existe dans la chaîne. Le procédé continue jusqu'à ce qu'on rencontre une lettre n'existant pas dans la chaîne. | LTRIM('ECOLE','EC') = |
| REPLACE( ch, |1,|2) | remplace toute occurrence |1 dans ch par |2. | REPLACE('SUCCEDE','SUC' 'PRE') = |
| RPAD(ch1,n[,ch2])  R = right pour droite | ch1 est suivi de ch2 sur une longueur totale de n positions | RPAD('UN',7,'-') = |
| RTRIM (Ch,[,chaîne])  ( R = right pour droite) | la dernière lettre de ch est enlevée si elle existe dans la chaîne. Le procédé continue jusqu'à ce qu'on rencontre une lettre n'existant pas dans la chaîne. | RTRIM('ECOLE','EC') = |
| SUBSTR(ch,m[,n]) | affiche n caractères de ch, commencant par le m ième caractère | SUBSTR('TRAVAIL',2,4)= |
| TRANSLATE(ch,ancienne,nouvelle) | remplace toute occurence ancienne de ch par nouvelle | SELECT poste, TRANSLATE(to\_char(sal),'5000','xxxx') salaire from emp  WHERE poste = 'president'; |
| UPPER(ch) | tous les caractères de ch sont en majuscule. | UPPER('exemple')= |

## FONCTIONS DE CARACTÈRES RETOURNANT DES NOMBRES

|  |  |  |
| --- | --- | --- |
| FONCTION  ch = chaîne de caractères | valeur retournée | Exemple |
| ASCII(ch) | valeur décimale du caractère | ASCII('Q') = |
| LENGTH(ch) | longueur de ch en caractères | LENGTH('ECOLE') = |
| INSTR(ch1,ch2¸[,N[,M]]) | Position de la m ième occurrence de ch2 dans ch1, commencant par le niène caractère. | INSTR('ECOLE','CO',1,1)=  INSTR('COCO','CO',1,2) =  INSTR('COICOI','OI',2,2)= |

## FONCTIONS DE TYPE DATE

Ces fonctions de date retournent des valeurs qui sont également de type DATE

|  |  |  |
| --- | --- | --- |
| Fonction | valeur retournée |  |
| ADD\_MONTHS  (date, n\_mois) | Ajoute n mois à la date et retourne une date comme résultat | SELECT ADD\_MONTHS (datembauche,12) from emp = |
| LAST\_DAY(date) | Dernier jour du mois qui contient la date | SELECT LAST\_DAY(SYSDATE) FROM DUAL= |
| NEXT\_DAY(date,*char*) | Calcule la date du premier jour de la semaine indiqué par *char* qui suit la date. *Char* doit correspondre à un jour de la semaine | SELECT NEXT\_DAY('03-01-16','LUNDI') FROM DUAL= |
| SYSDATE | date courante |  |
| ROUND(date,[,précision]) | La date est arrondie jusqu'à la précision spécifiée | SELECT ROUND(TO\_DATE('03-01-28'),'YEAR') FROM DUAL  = |
| TRUNC(date,[,*format*]) | La date est tronquée à la précision spécifiée par *format* | SELECT TRUNC(TO\_DATE('03-02-10'),'DAY') FROM DUAL = |