

## Casos Prácticos TEMA 4

### 1.- Tabla resumen formulas.

ABS - SQL> SELECT APELLIDO, SALARIO, ABS(SALARIO-10000) FROM EMPLE;

CEIL - SQL> SELECT CEIL(20.3), CEIL(16), CEIL(-20.3), CEIL(-16) FROM DUAL;

FLOOR - SQL> SELECT FLOOR(20.3), FLOOR(16), FLOOR(-20.3), FLOOR(-16) FROM DUAL;

MOD - SQL> SEELCT MOD(11.4), MOD(10, -15), MOD(-10, -3), MOD(10.4, 4.5) FROM DUAL;

NVL - SQL> SELECT SALARIO, COMISION, SALARIO + COMISION FROM EMPLE;

POWER - SQL> SELECT POWER(2,4), POWER(2, -4), POWER(3.5, 2.4), POWER(4.5, 2) FROM DUAL;

ROUND - SQL> SELECT ROUND(1.56, 1), ROUND(1.56), ROUND(1.2234, 2), ROUND(1.2676, 3) FROM DUAL;

- SQL> SELECT ROUND(145.5, -1), ROUND(145.5, -2), ROUND(145.5, -3), ROUND(141, -1), ROUND(145, -1) FROM DUAL;

SIGN - SQL> SELECT SIGN(-10), SIGN(10) FROM DUAL;

SORT - SQL> SELECT SORT(25), SORT(25.6) FROM DUAL;

TRUNC - POSITIVO: SQL> SELECT TRUNC(1.5634, 1), TRUNC(1.1684, 2), TRUNC(1.662) FROM DUAL;

- NEGATIVO: SQL> SELECT TRUNC(187.98, -1), TRUNC(187.98, -2), TRUNC(187.98, -3) FROM DUAL;

2.- AVG - SQL> SELECT AVG(SALARIO) FROM EMPLE WHERE DEPT\_NO = 10;

COUNT - SQL> SELECT COUNT(\*) FROM EMPLE;

MAX - SQL> SELECT MAX(SALARIO) FROM EMPLE;  
SQL> SEELCT MAX(APELLIDO) FROM EMPLE;  
SQL> SELECT APELLIDO, SALARIO FROM EMPLE WHERE SALARIO = (SELECT MAX(SALARIO) FROM EMPLE);

MIN - SQL> SELECT MIN(SALARIO) FROM EMPLE;  
SQL> SELECT APELLIDO, SALARIO FROM EMPLE WHERE SALARIO = (SELECT MIN(SALARIO) FROM EMPLE);

SUM - SQL> SELECT SUM(SALARIO) FROM EMPLE;

```
VARIANCE - SQL> SELECT VARIANCE(SALARIO) FROM EMPLE;
```

```
3.- SQL> SELECT COUNT(OFIGICIO) "OFICIOS" FROM EMPLE;
```

```
4.- SQL> SELECT NOMBRE_ALUMNO, GREATEST(NOTA1, NOTA2, NOTA3) "MAYOR",  
LEAST(NOTA1, NOTA2, NOTA3) "MENOR" FROM NOTAS_ALUMNOS;  
SQL> SELECT GREATEST('BENITO', 'JORGE', 'ANDRÉS', 'ISABEL') FROM DUAL;
```

```
5.- SQL> SELECT CHR(75), CHR(65) FROM DUAL;
```

```
SQL> SELECT CONCAT('El apellido es: ', APELLIDO) FROM EMPLE;  
SQL> SELECT LOWER('oRACle Y sqL') "minuscúla", UPPER('oRACle Y sqL')  
"MAYUSCULA", INITCAP('oRACle y sqL') "Tipo Título" FROM DUAL;  
SQL> SELECT LPAD(NOMBRE_ALUMNO, 30, '.') "IZQUIERDA", RPAD(NOMBRE_ALUMNO,  
30, '.') "DERECHA" FROM NOTAS_ALUMNOS;  
SQL> SELECT LTRIM('          hola') || RTRIM('          adios  ') || '*'  
FROM DUAL;  
SQL> SELECT * FROM MISTEXTOS  
SQL> SELECT RTRIM (TITULO, '.' ) FROM MISTEXTOS;  
SQL> SELECT LTRIM (TITULO, '.' ) FROM MISTEXTOS;  
SQL> SELECT REPLACE('BLANCO Y NEGRO', 'Ø', 'AS') FROM DUAL;  
SQL> SELECT REPLACE('BLANCO Y NEGRO', 'Ø') FROM DUAL;  
SQL> SELECT SUBSTR('ABCDEFGH', 3,2) "s1", SUBSTR ('ABCDEFGH', -3,2) "s2",  
SUBSTR('ABCDEFGH', 4) "s3" FROM DUAL;  
SQL> SELECT APELLIDO, SUBSTR(APELLIDO,1,1) FROM EMPLE WHERE DEPT_NO = 10;  
SQL> SELECT TRANSLATE('LOS PILARES DE LA TIERRA', 'AEIOU', 'aeiou') FROM  
DUAL;
```

```
SQL> SELECT TRANSLATE('LOS PILARES DE LA TIERRA', 'LAEIOU', '1') FROM  
DUAL;
```

```
6.- SQL> SELECT ASCII ('A') FROM DUAL
```

Si ponemos como argumento una cadena de caracteres, visualiza sólo el valor ASCII del primer carácter de la cadena.

**INSTR**(cad1, cad2 [,comienzo [,m] ] ). A partir de la cadena 'II VUELTA CICLISTA A TALAVERA' encuentra la segunda ocurrencia 'TA' desde la posición 3:

```
SQL> SELECT INSTR('II VUELTA CICLISTA A TALAVERA', 'TA', 3, 2) "EJEMPLO"  
FROM DUAL;
```

EJEMPLO

17

Cuando «comienzo» es negativo (-1), se comienza la búsqueda en la posición final y se va de derecha a izquierda en la cadena. Devuelve la posición contando desde la izquierda, es decir, la primera A que encuentra desde la primera posición empezando por el final:

```
SQL> SELECT INSTR('II VUELTA CICLISTA A TALAVERA', 'A', -1) "EJEMPLO"  
FROM DUAL;
```

## EJEMPLO

29

Si en el ejemplo anterior empleamos un -2, haría que la función comenzase desde la segunda posición empezando por el final; un -3 haría que se iniciase desde la tercera posición y, así, sucesivamente.

A partir de la tabla MISTEXTOS, encuentra la posición de la segunda ocurrencia de la letra 'A' en la columna AUTOR a partir del comienzo:

```
SQL> SELECT AUTOR, INSTR(AUTOR, 'A', 1, 2) FROM MISTEXTOS;
```

AUTOR	INSTR(AUTOR, 'A', 1, 2)
ALCALDE GARCÍA	4
GARCÍA GARCERÁN	6
GARCÍA STRUCH	6
RUÍZ LOPEZ	0
RUÍZ LOPEZ	0

Si en la función INSTR, "cad2" es un conjunto de caracteres, entonces la función devuelve la posición donde comienza el primer carácter de ese conjunto.

**LENGTH(cad).** Calcula el número de caracteres de las columnas TITULO y AUTOR para todas las filas de la tabla MISTEXTOS:

```
SQL> SELECT TITULO, LENGTH(TITULO), AUTOR, LENGTH(AUTOR) FROM MISTEXTOS;
```

TITULO	LENGTH(TITULO)	AUTOR	LENGTH(AUTOR)
METODOLOGÍA DE LA PROGRAMACIÓN.	31	ALCALDE GARCÍA	14
"INFORMÁTICA BÁSICA."	21	GARCÍA GARCERÁN	15
SISTEMAS OPERATIVOS	19	GARCÍA STRUCH	13
SISTEMAS DIGITALES.	19	RUÍZ LOPEZ	10
"MANUAL DE C."	14	RUÍZ LOPEZ	10

**7.- SYSDATE.** Esta función devuelve la fecha del sistema. Por ejemplo:

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

```
SYSDATE
-----
03/08/05
```

**ADD\_MONTHS(fecha, n).** A partir de la tabla EMPLE, suma doce meses a la fecha de alta para los empleados del departamento 10:

```
SQL> SELECT FECHA_ALT, ADD_MONTHS(FECHA_ALT, 12) FROM EMPLE WHERE DEPT_NO = 10;
```

FECHA_AL	ADD_MONT
09/06/91	09/06/92
17/11/91	17/11/92
23/01/92	23/01/93

**LAST\_DAY(fecha).** Obtén de la tabla EMPLE el último día del mes para cada una de las fechas de alta de los empleados del departamento 10:

```
SQL> SELECT FECHA_ALT, LAST_DAY(FECHA_ALT) FROM EMPLE WHERE DEPT_NO = 10;
```

FECHA_AL	LAST_DAY
-----	-----
09/06/91	30/06/91
17/11/91	30/11/91
23/01/92	31/01/92

**MONTHS\_BETWEEN(fecha1, fecha2).** Cálculo de la edad: necesitamos la función "SYSDATE", que devuelve la fecha actual (fecha del sistema) y calculamos los meses transcurridos entre la fecha de hoy y la fecha de nacimiento. Dividimos entre 12 ese resultado y aplicamos la función TRUNC para suprimir decimales:

```
SQL> SELECT TRUNC (MONTHS_BETWEEN (SYSDATE, '18/11/1964') / 12) "Edad  
actual" FROM DUAL;
```

Edad actual
-----
40

**NEXT\_DAY(fecha, cad).** Si hoy es jueves 19 de octubre de 2006 (fecha del sistema «sysdate»), ¿qué fecha será el próximo jueves?

```
SQL> SELECT NEXT_DAY(SYSDATE, 'JUEVES') "Sig. Jueves" FROM DUAL;
```

Sig. Jue
-----
26/10/06

**8.-** A partir de la tabla EMPLE, obtén la fecha de alta (columna FECHA\_ALT) formateada, de manera que aparezca el nombre del mes con todas sus letras (month), el número de día de mes (dd) y el año (yyyy), para aquellos empleados del departamento 10:

```
SQL> SELECT TO_CHAR (FECHA_ALT, 'month DD, YYYY') "NUEVA FECHA" FROM  
EMPLE WHERE DEPT_NO = 10;
```

NUEVA	FECHA
-----	-----
junio	09, 1991
noviembre	17, 1991
enero	23, 1992

Ahora se desea obtener la fecha de alta de forma que aparezca el nombre del mes con tres letras (mon), el número de día del año (ddd), el último dígito del año (y) y los tres últimos dígitos del año (yyy):

```
SQL> SELECT TO_CHAR (FECHA_ALT, 'mon ddd y yyy') "FECHA" FROM EMPLE WHERE  
DEPT_NO = 10;
```

FECHA
-----
jun 160 1 991
nov 321 1 991
ene 023 2 992

## 8.2.-

Podemos cambiar el valor por omisión para la fecha con el parámetro `NLS_DATE_FORMAT`, usando la orden `ALTER SESSION`. Por ejemplo, para cambiar el formato de la fecha y que aparezca de la siguiente manera: día/nombre mes/ año hora:minutos:segundos utilizaremos la siguiente sentencia:

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

Sesión modificada.

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

```
SYSDATE
-----
03/agosto   /2005 17:20:48
```

## 9.-

Por defecto, al iniciar nuestra sesión el idioma definido para la fecha es el español. Podemos definir otro idioma, por ejemplo, el francés, de la siguiente manera: `SQL> ALTER SESSION SET NLS_DATE_LANGUAGE=French;`

Si queremos visualizar el nombre del mes y el día de la semana, éstos aparecerán en francés, con lo que la fecha de hoy nos quedará así: `SQL> SELECT TO_CHAR(sysdate, 'Hoy es " day ", " dd " de " month " de " yyyy' )`

```
"Fecha" FROM DUAL;
```

```
Fecha
-----
Hoy es mercredi , 03   de août           de 2005
```

**9.2.-** -9- `SQL> SELECT TO_CHAR(1, '999'), TO_CHAR(-1, '999'), TO_CHAR(01, '999'), TO_CHAR(01, '999') , TO_CHAR(0, '999') FROM DUAL;`

-0- `SQL> SELECT TO_CHAR(10, '0999'), TO_CHAR(10, '9990'), TO_CHAR(10, '990090') FROM DUAL;`

-\$- `SQL> SELECT TO_CHAR(10, '$9999') , TO_CHAR(10, '$009'), TO_CHAR(10, '99$') FROM DUAL;`

-B- `SQL> SELECT TO_CHAR(0, 'B999'), TO_CHAR(5, 'B999') FROM DUAL;`

-MI- `SQL> SELECT TO_CHAR(-55, '999MI'), TO_CHAR(55, '999MI') FROM DUAL;`

-S- `SQL> SELECT TO_CHAR(-55, '999S'), TO_CHAR(-55, 'S999'), TO_CHAR(55, 'S999'), TO_CHAR(55, '999S') FROM DUAL;`

**9.3.-** `SQL> ALTER SESSION SET NLS_NUMERIC_CHARACTERS='/*'`

Sesión modificada.

```
SQL> SELECT TO_CHAR(12345.67, '999G999D999') FROM DUAL;
```

```
SQL> ALTER SESSION SET NLS_NUMERIC_CHARACTERS=',.';
```

```
SQL> ALTER SESSION SET NLS_CURRENCY='PESETAS';
```

Sesión modificada.

```
SQL> SELECT TO_CHAR(123, 'L999') FROM DUAL;
```

```
10.- SQL> SELECT TO_NUMBER('-123456') "NUMERO1", TO_NUMBER('123,99',  
'999D99') "NUMERO2" FROM DUAL;
```

```
SQL> SELECT TO_NUMBER('123.456', '999G999') "NO CONVIERTE" FROM DUAL;
```

```
NO CONVIERTE  
-----  
          123456
```

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT='DD/MM/YYYY
```

```
SQL> SELECT TO_DATE('01012006') FROM DUAL
```

```
SELECT TO_DATE('010106') FROM DUAL
```

```
SQL> SELECT TO_DATE('01012006') FROM DUAL;
```

```
SQL> SELECT TO_CHAR(TO_DATE('01012007', 'ddmmyyyy'), 'Month') "MES" FROM  
DUAL;
```

```
11.- SQL> SELECT APELLIDO, OFICIO, DECODE(UPPER(OFICIO), 'PRESIDENTE', 1,  
'EMPLEADO', 2, 5) "Codigo" FROM EMPL;
```

```
SQL> SELECT APELLIDO, VSIZE(APELLIDO) BYTES FOM EMPL WHERE DEPT_NO=10;
```

```
SQL> SELECT APELLIDO, DUMP(APELLIDO, 16, 1, 4) FROM EMPL WHERE APELLIDO  
LIKE 'SALA';
```

```
SQL> SELECT USER, UID FROM DUAL;
```

```
SQL> SHOW USER;
```