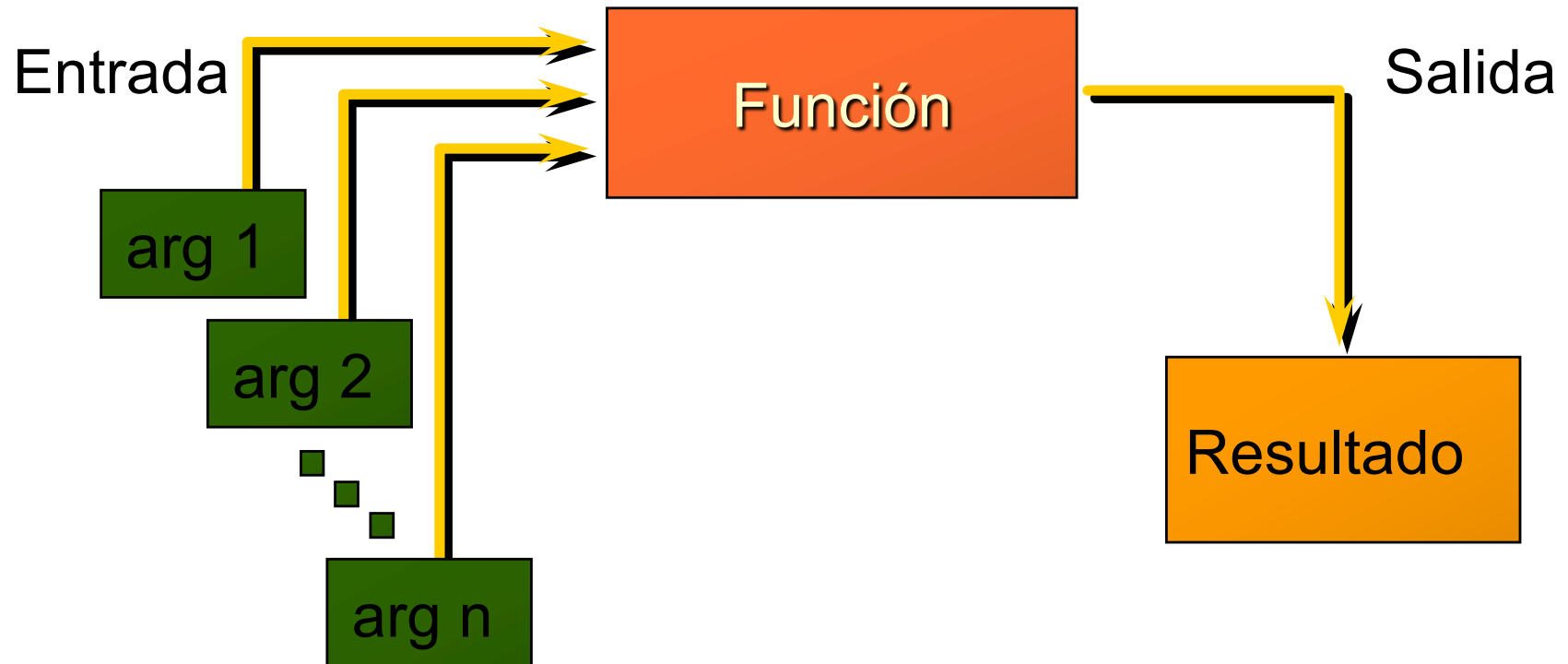


SQL

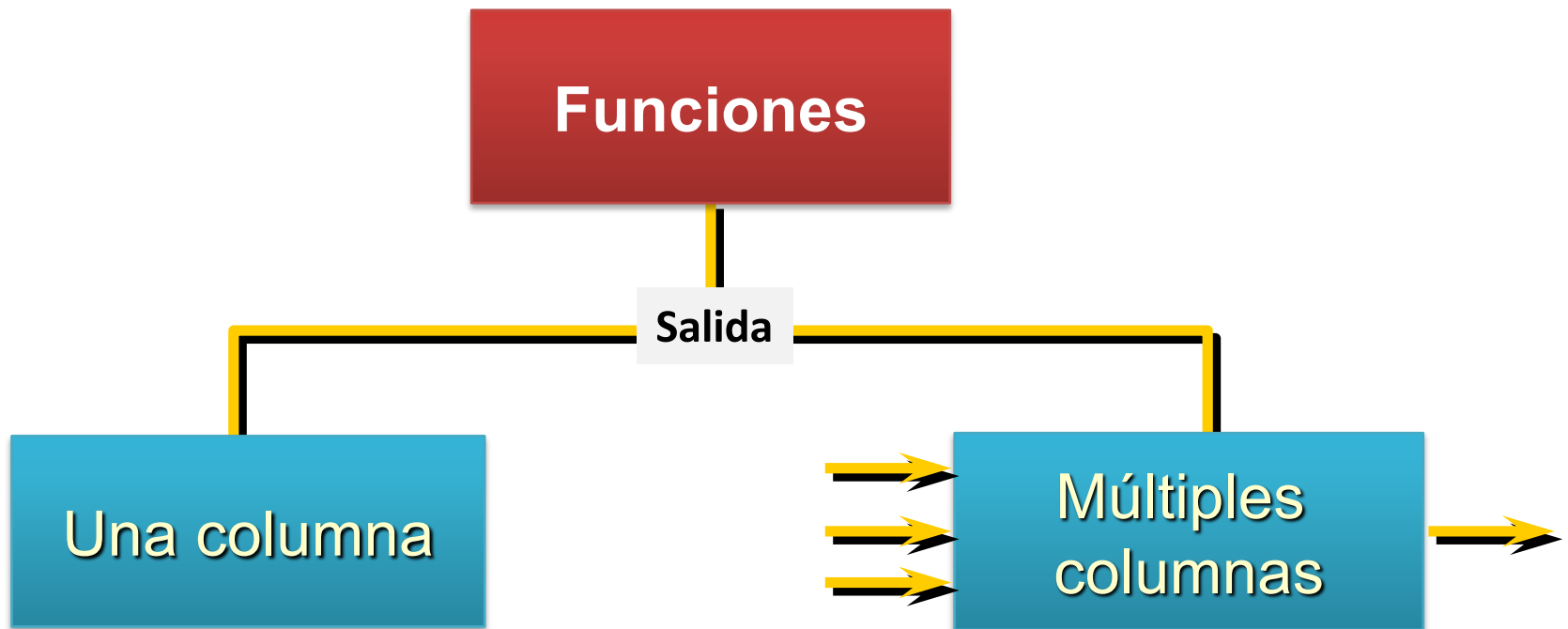


Funciones

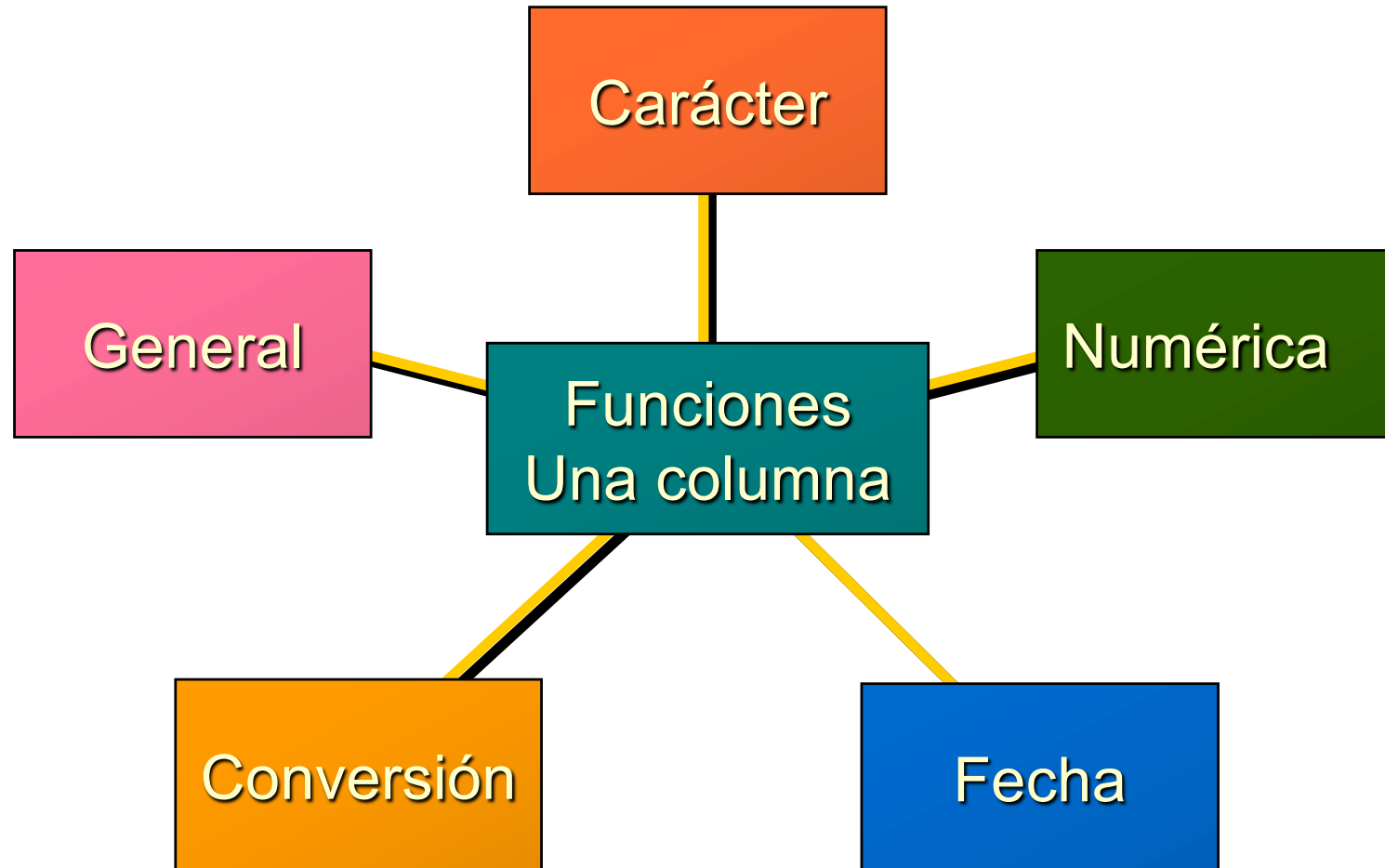
Funciones



Funciones

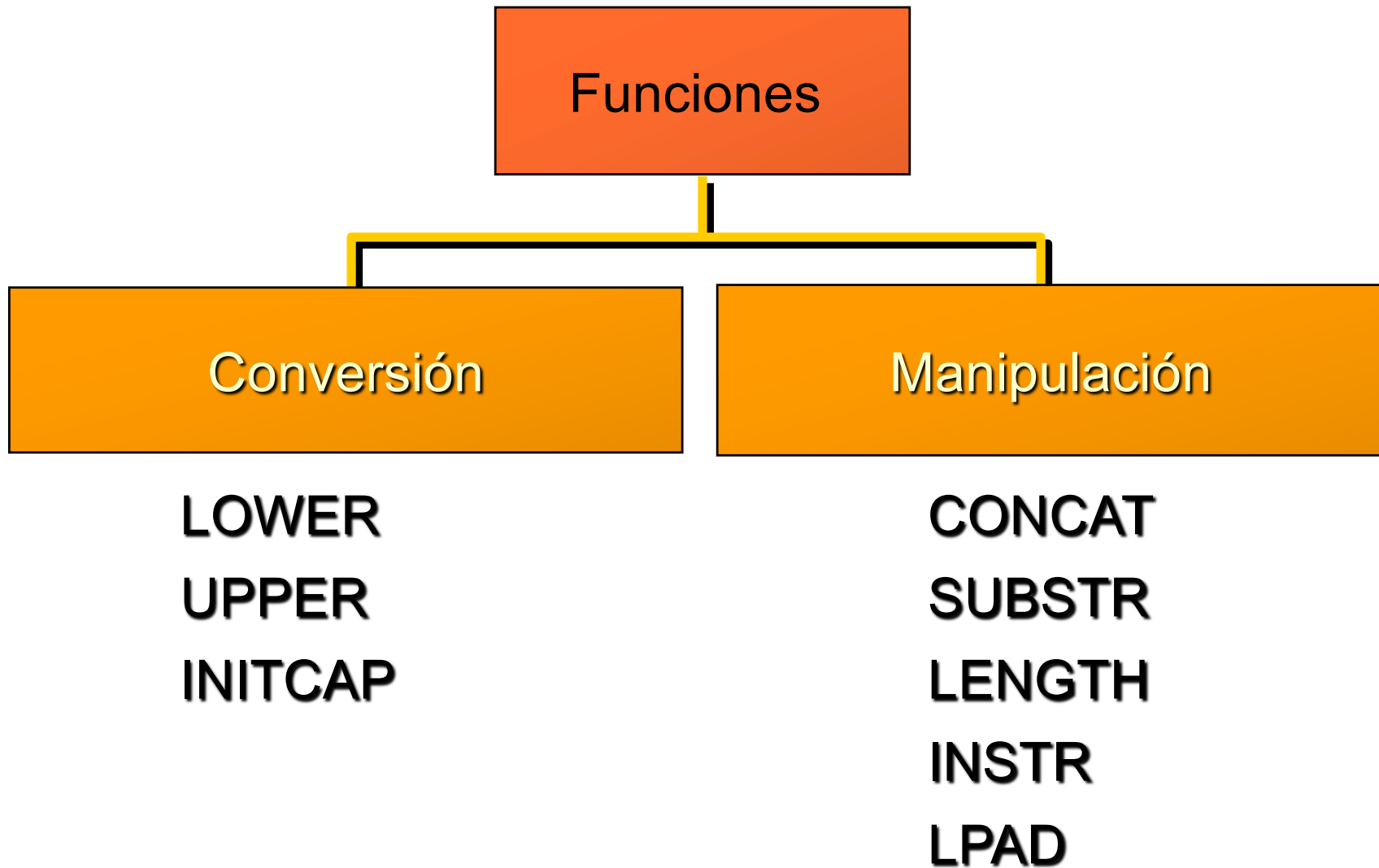


Una columna



```
function_name (column|expression, [arg1, arg2,...])
```

Cadenas de Caracteres



Cadenas de Caracteres

Función	Resultado
LOWER(' SQL curso ')	sql course
UPPER(' SQL curso ')	SQL COURSE
INITCAP(' SQL curso ')	Sql Course

Cadenas de Caracteres

```
SQL> SELECT  empno, ename, deptno  
2  FROM      emp  
3  WHERE      ename = 'blake';
```

no rows selected

```
SQL> SELECT  empno, ename, deptno  
2  FROM      emp  
3  WHERE      LOWER(ename) = 'blake';
```

EMPNO	ENAME	DEPTNO
7698	BLAKE	30

Cadenas de Caracteres

Función	Resultado
CONCAT('Una ' , ' cadena ')	Una cadena
SUBSTR(' String ' ,1,3)	Str
LENGTH(' String ')	6
INSTR(' String ' , ' r ')	3
LPAD(sal,10, ' * ')	*****5000

Funciones para manipular cadenas de caracteres

```
SQL> SELECT  ename, CONCAT (ename, job), LENGTH(ename),  
2          INSTR(ename, 'A')  
3 FROM      emp  
4 WHERE     SUBSTR(job, 1, 5) = 'SALES';
```

ENAME	CONCAT (ENAME, JOB)	LENGTH (ENAME)	INSTR (ENAME, 'A')
MARTIN	MARTINSALESMAN	6	2
ALLEN	ALLENSALESMAN	5	1
TURNER	TURNERSALESMAN	6	0
WARD	WARDSALESMAN	4	2

Funciones numéricas

- ROUND (Redondea el valor)

ROUND(45.926, 2) 45.93

- TRUNC (Trunca el valor)

TRUNC(45.926, 2) 45.92

- MOD (Regresa el residuo de la división)

%

MOD(1600, 300) 100

1600 % 300 100

Función ROUND

```
SQL> SELECT ROUND(45.923, 2), ROUND(45.923, 0),  
2         ROUND(45.923, -1)  
3 FROM DUAL;
```

ROUND(45.923, 2)	ROUND(45.923, 0)	ROUND(45.923, -1)
----- 45.92	----- 46	----- 50

Función TRUNC

```
SQL> SELECT TRUNC (45.923, 2), TRUNC (45.923),  
2          TRUNC (45.923, -1)  
3 FROM DUAL;
```

TRUNC (45.923, 2)	TRUNC (45.923)	TRUNC (45.923, -1)	
-----	-----	-----	
45.92	45	40	

Función MOD

```
SQL> SELECT  ename, sal, comm, MOD(sal, comm)
2 FROM      emp
3 WHERE     job = 'SALESMAN';
```

ENAME	SAL	COMM	MOD (SAL, COMM)
-----	-----	-----	-----
MARTIN	1250	1400	1250
ALLEN	1600	300	100
TURNER	1500	0	1500
WARD	1250	500	250

- SYSDATE es una función que devuelve la fecha y hora
- DUAL es una tabla temporal empleada para ver el resultado de sysdate

- Sumar o restar un número a partir de una fecha o de un valor de fecha resultante.
- Restar dos fechas para encontrar el número de días entre esas fechas.
- Añadir horas a una fecha dividiendo el número de horas por 24.

Operaciones Aritméticas con campos tipo Fecha

```
SQL> SELECT  ename, (SYSDATE-hiredate)/7 WEEKS  
2    FROM    emp  
3    WHERE   deptno = 10;
```

ENAME	WEEKS
KING	830.93709
CLARK	853.93709
MILLER	821.36566

Funciones con Fechas

FUNCIÓN	DESCRIPCIÓN
MONTHS_BETWEEN	Número de meses entre dos fechas
ADD_MONTHS	Adiciona meses calendario a la fecha
NEXT_DAY	Siguierte día de la fecha solicitada
LAST_DAY	Ultimo día del mes
ROUND	Redondea la fecha
TRUNC	Trunca la fecha

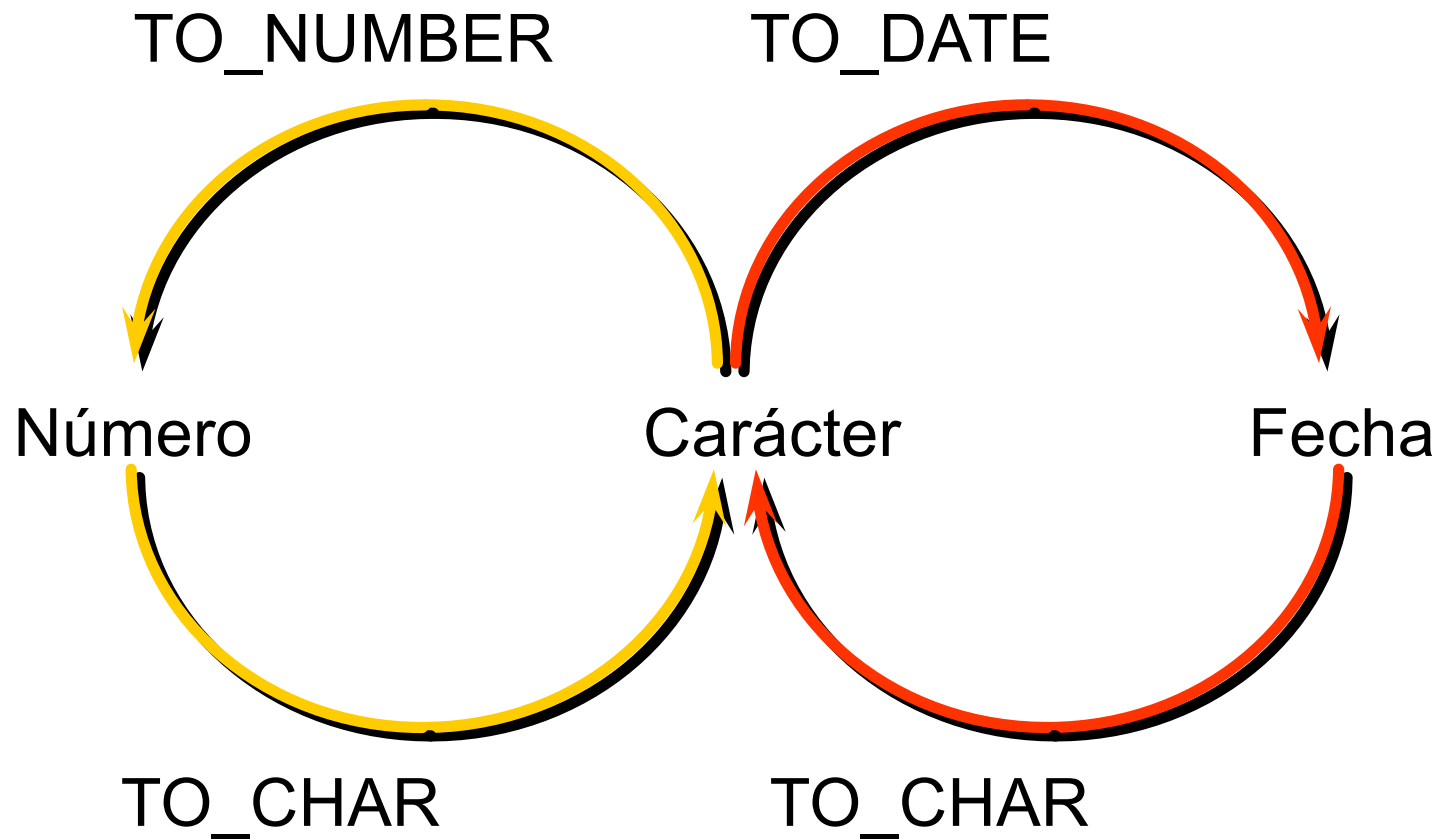
Funciones con Campos Tipo Fecha

- MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')
➡ 19.6774194
- ADD_MONTHS ('11-JAN-94',6) ➡ '11-JUL-94'
- NEXT_DAY ('01-SEP-95','FRIDAY') ➡ '08-SEP-95'
- LAST_DAY('01-SEP-95') ➡ '30-SEP-95'

Funciones con Campos Tipo Fecha

- **ROUND('25-JUL-95','MONTH') ➡ 01-AUG-95**
- **ROUND('25-JUL-95','YEAR') ➡ 01-JAN-96**
- **TRUNC('25-JUL-95','MONTH') ➡ 01-JUL-95**
- **TRUNC('25-JUL-95','YEAR') ➡ 01-JAN-95**

Conversión de Tipo de Datos



Elementos

YYYY	Año con los 4 dígitos
MM	2 dígitos para el mes
MONTH	Nombre del Mes
DY	La abreviación del día de la semana (3 caracteres)
DAY	Nombre del día

TO_CHAR

```
SQL> SELECT ename,  
2         TO_CHAR(hiredate, 'fmDD Month YYYY') HIREDATE  
3 FROM emp;
```

ENAME	HIREDATE
-----	-----
KING	17 November 1981
BLAKE	1 May 1981
CLARK	9 June 1981
JONES	2 April 1981
MARTIN	28 September 1981
ALLEN	20 February 1981
...	

14 rows selected.

TO_CHAR

```
TO_CHAR(number, 'fmt')
```

Fmt: Formato

9	Representa un número
0	Fuerza a desplegar un cero
\$	Coloca un signo flotante \$
L	Usa el símbolo de moneda parametrizado
.	Imprime el punto decimal
,	Imprime el indicador de miles

TO_CHAR

```
SQL> SELECT TO_CHAR(sal, '$99,999') SALARY  
2 FROM emp  
3 WHERE ename = 'SCOTT';
```

SALARY

\$3,000

Convierte un valor nulo a un determinado valor

Puede utilizar los tipos de datos fecha, el carácter y número.

El tipo de datos de los argumentos debe ser el mismo:

NVL (comm, 0)

NVL (HireDate, '01-JAN-97 ')

NVL (trabajo, 'No trabaja')

Función NVL

```
SQL> SELECT ename, sal, comm, (sal*12)+NVL(comm,0)
       2 FROM emp;
```

ENAME	SAL	COMM	(SAL*12)+NVL(COMM,0)
KING	5000		60000
BLAKE	2850		34200
CLARK	2450		29400
JONES	2975		35700
MARTIN	1250	1400	16400
ALLEN	1600	300	19500
...			

14 rows selected.

Función DECODE

```
SQL> SELECT job, sal,  
2          DECODE(job, 'ANALYST', SAL*1.1,  
3                    'CLERK',   SAL*1.15,  
4                    'MANAGER', SAL*1.20,  
5                               SAL)  
6          REVISED_SALARY  
7 FROM emp;
```

JOB	SAL	REVISED_SALARY
PRESIDENT	5000	5000
MANAGER	2850	3420
MANAGER	2450	2940
...		

14 rows selected.

Función DECODE

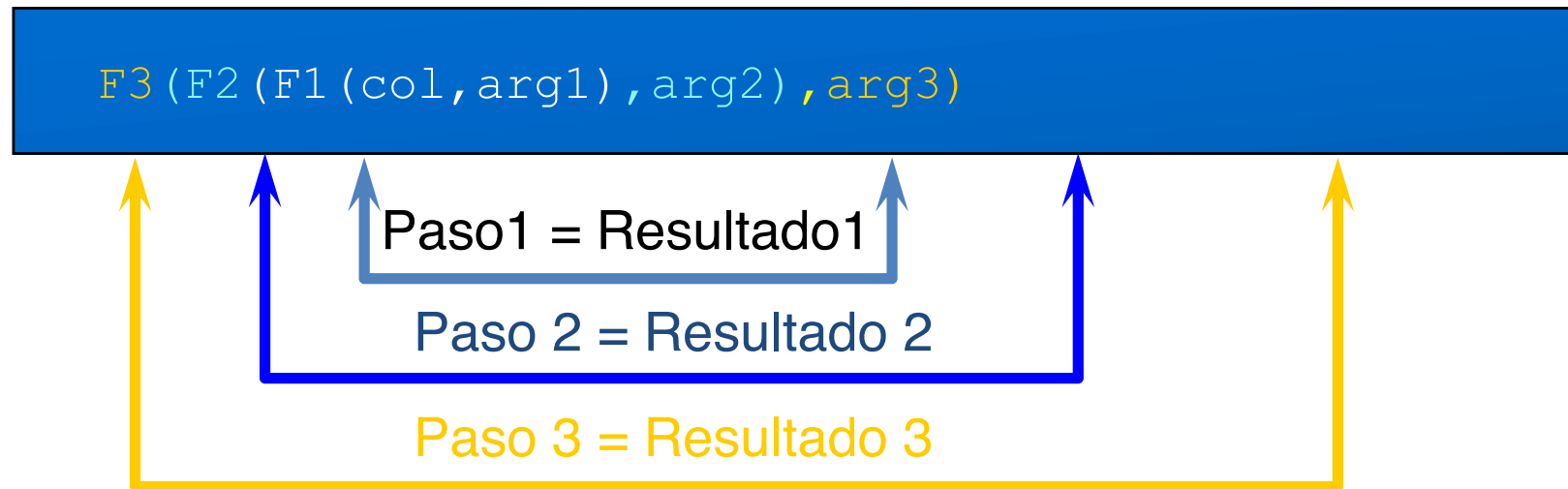
```
SQL> SELECT job, sal,  
2         IF job = 'ANALYST' THEN SAL*1.1  
3         ELSE IF JOB = 'CLERK' THEN SAL*1.15  
4         ELSE IF JOB = 'MANAGER' THEN SAL*1.20  
5         ELSE SAL END  
6         REVISSED_SALARY  
7 FROM emp;
```

JOB	SAL	REVISSED_SALARY
PRESIDENT	5000	5000
MANAGER	2850	3420
MANAGER	2450	2940
...		

14 rows selected.

Funciones Anidadas

- Las funciones se pueden anidar hasta cualquier nivel.
- Las funciones anidadas se evalúan de nivel más profundo el nivel menos profundo



Funciones Anidadas

```
SQL> SELECT      ename,  
2              NVL (TO_CHAR (mgr) , 'No Manager')  
3  FROM          emp  
4  WHERE         mgr IS NULL;
```

ENAME	NVL (TO_CHAR (MGR) , 'NOMANAGER')
-----	-----
KING	No Manager