

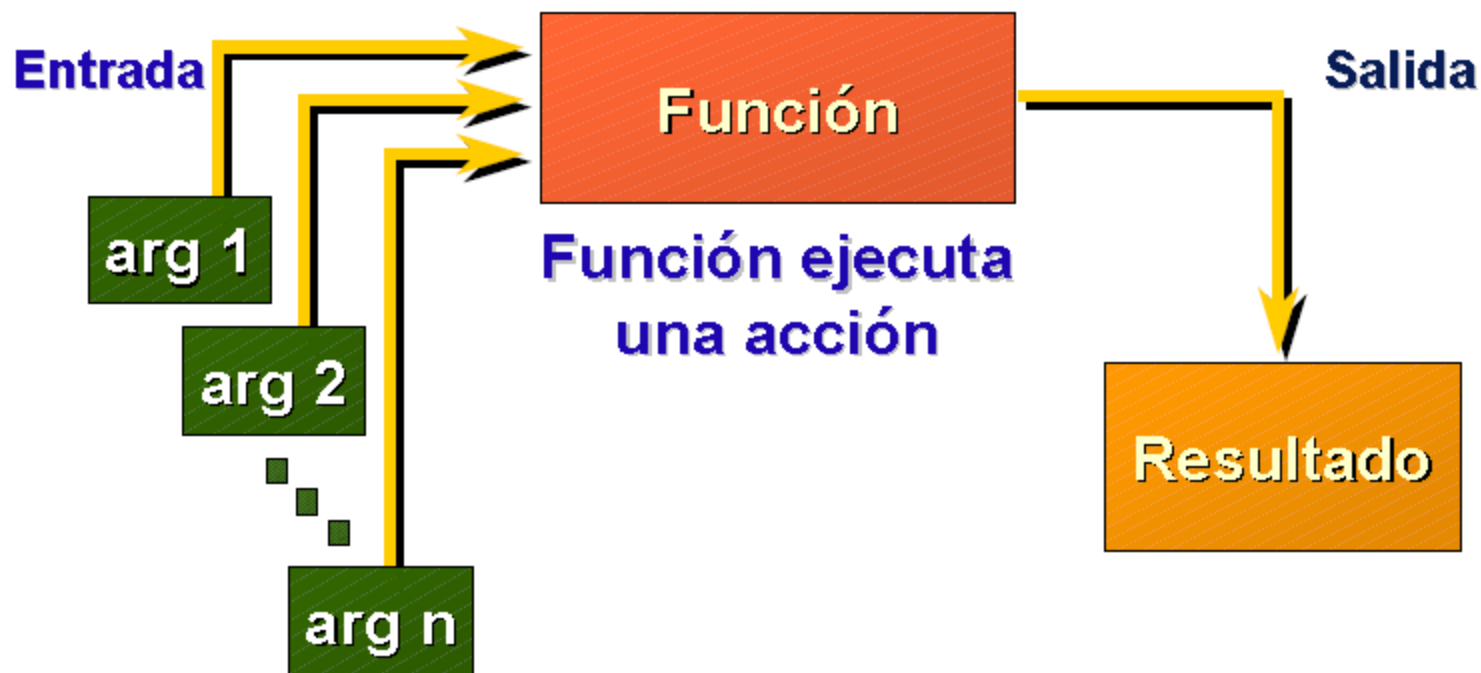
Parte 4

Funciones de Renglón Simples

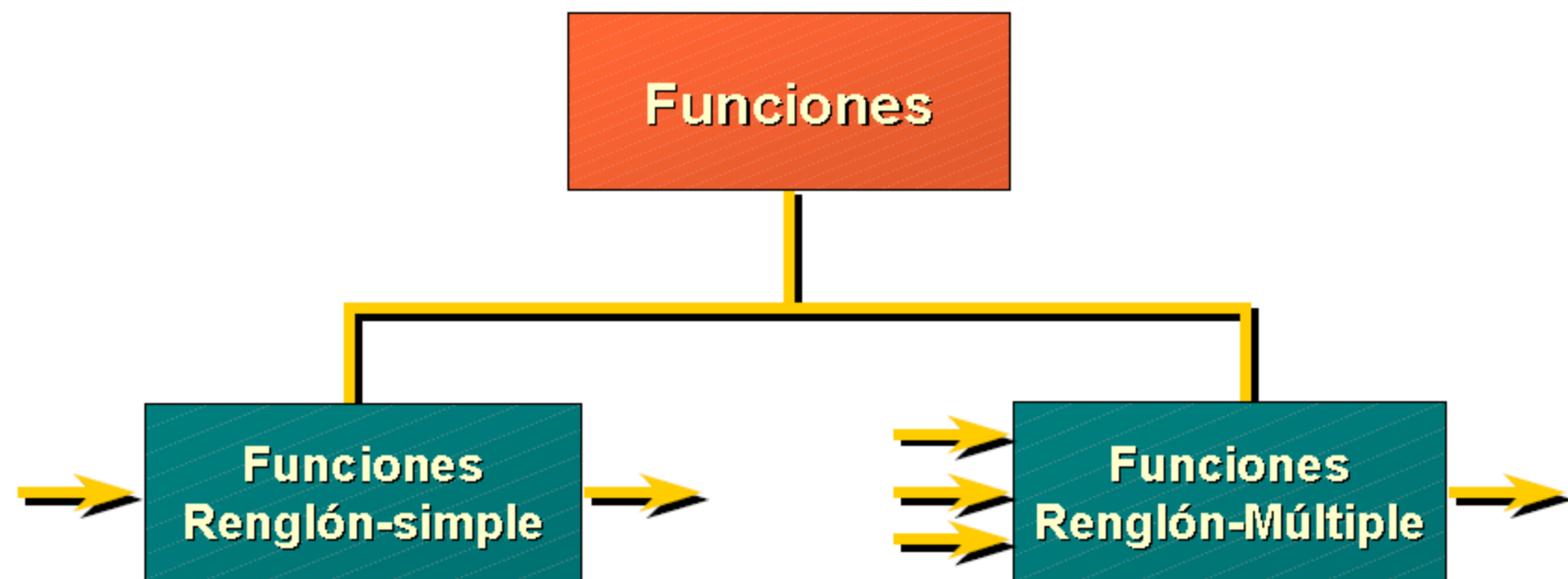
Objetivos

- Al completar esta lección, deberá ser capaz de hacer lo siguiente :
 - Describir los diferentes tipos de funciones disponibles en SQL
 - Utilizar funciones para caracteres, números y fechas en la instrucción SELECT

Funciones de SQL



Tipos de Funciones de SQL

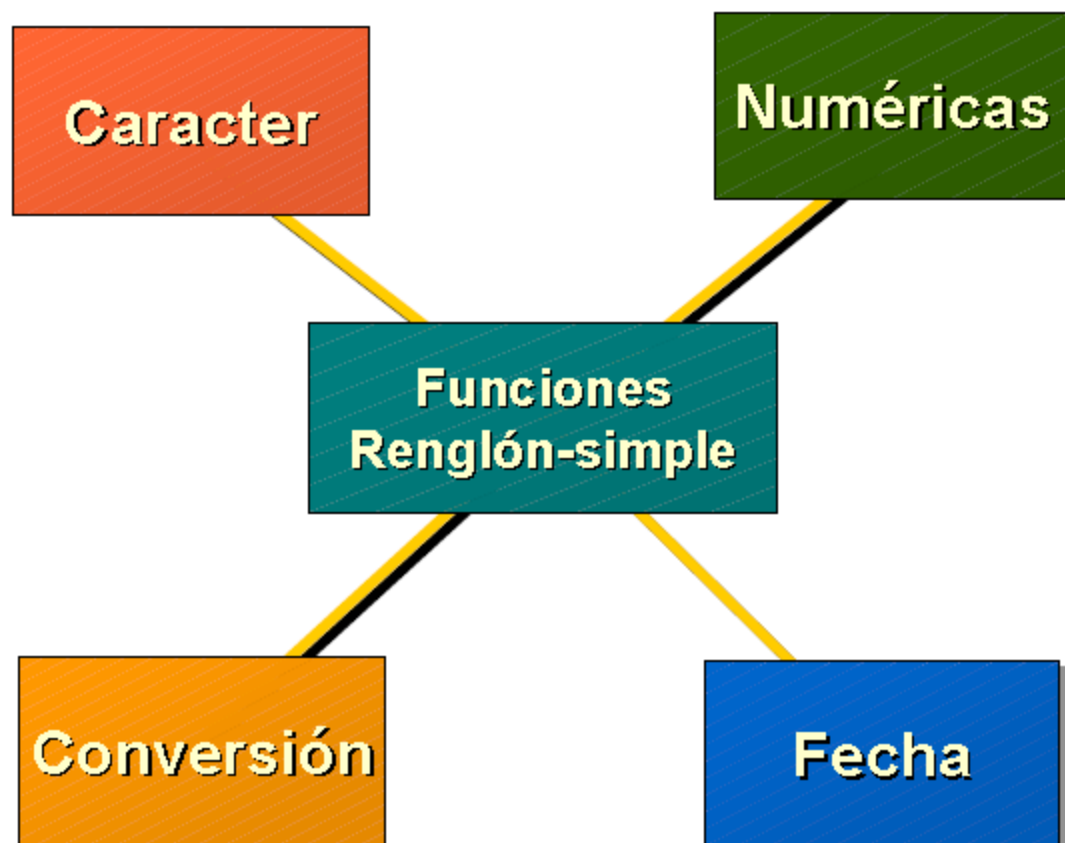


Funciones de Renglón-simple

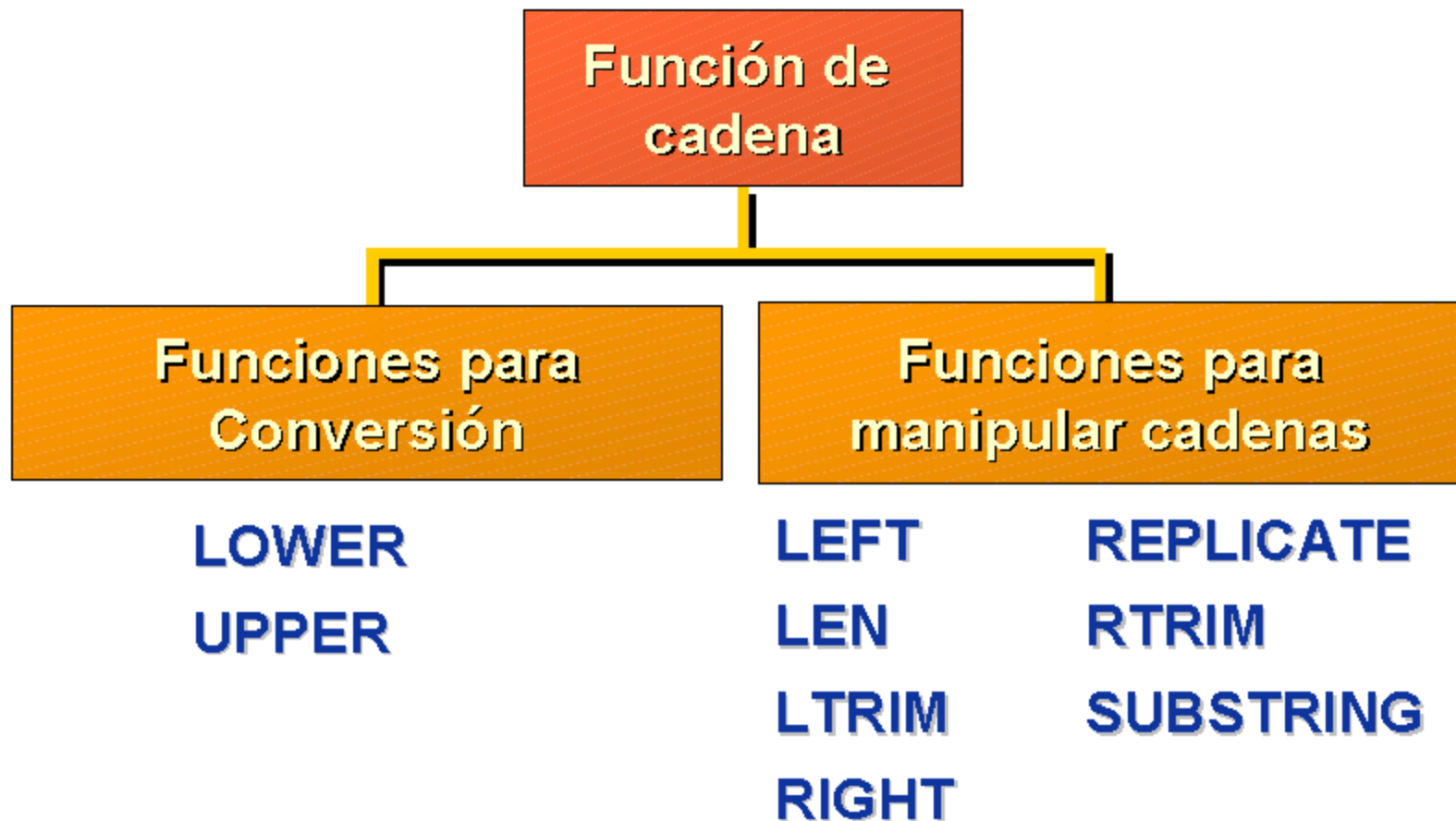
- Manipulan datos
- Aceptan argumentos y retornan un valor
- Actúan sobre cada renglón
- Retornan un resultado por renglón
- Pueden ser anidadas

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

Funciones de Renglón-simple



Funciones para cadenas



Funciones para Conversión

| Función | Resultado |
|-------------------------|------------|
| LOWER(' SQL Course ') | sql course |
| UPPER(' SQL Course ') | SQL COURSE |

Funciones para manipular cadenas

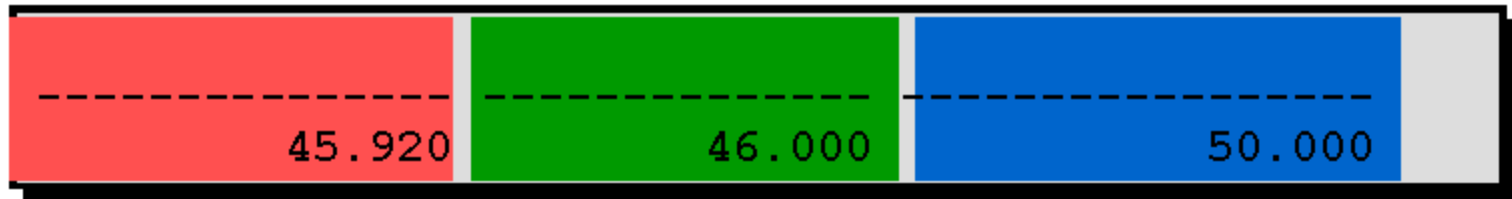
| Función | Resultado |
|-----------------------------|-----------|
| LEFT(' Good ', 2) | Go |
| SUBSTRING(' String ', 1, 3) | Str |
| LEN(' String ') | 6 |
| LTRIM(' String ') | String |
| RIGHT(' Good ', 2) | od |
| RTRIM(' String ') | String |
| REPLICATE(' S' , 4) | SSSS |

Funciones numéricas

- **ROUND:** Redondea un valor a las posiciones decimales indicadas.
- **POWER:** Eleva una cantidad a la potencia indicada.
- **ABS:** Retorna el valor absoluto de una expresión .

Utilizando la Función ROUND

```
SELECT ROUND(45.923, 2), ROUND(45.923,0), ROUND(45.923, -1);
```



Utilizando ROUND como una Función TRUNC

```
SELECT ROUND(45.923,2,1), ROUND(45.923,0,1),  
ROUND(45.923,-1,1)
```

45.920

45.000

40.000

Utilizando la Función POWER

Obtenga la raíz cuadrada del salario de todos los empleados cuyo puesto sea "SALESMAN".

```
SELECT  ename, sal, comm, POWER(sal, 0.5)
FROM    emp
WHERE   job = 'SALESMAN' ;
```

| ENAME | SAL | COMM | |
|--------|---------|---------|-------|
| ----- | ----- | ----- | ----- |
| ALLEN | 1600.00 | 300.00 | 40.00 |
| WARD | 1250.00 | 500.00 | 35.36 |
| MARTIN | 1250.00 | 1400.00 | 35.36 |
| TURNER | 1500.00 | .00 | 38.73 |

Utilizando la Función ABS

Obtenga cuantas unidades hay entre el salario y la comisión para los empleados cuyo puesto sea "SALESMAN".

```
SELECT  ename, sal, comm, ABS (sal - comm)
FROM    emp
WHERE   job = 'SALESMAN' ;
```

| ENAME | SAL | COMM | |
|--------|---------|---------|---------|
| ALLEN | 1600.00 | 300.00 | 1300.00 |
| WARD | 1250.00 | 500.00 | 750.00 |
| MARTIN | 1250.00 | 1400.00 | 150.00 |
| TURNER | 1500.00 | .00 | 1500.00 |

Trabajando con Fechas

- **SQL Server almacena las fechas en un formato numérico internamente que representa : Año, Mes, Día, Hora, Minutos, Segundos y milésimas de segundo.**
- **GETDATE() es una función que obtiene la fecha actual del sistema.**

Funciones para Fechas

| FUNCION | DESCRIPCION |
|-----------------|--|
| DATEADD | Agrega intervalo a la fecha indicada |
| DATEDIFF | Devuelve el número de <i>unidades</i> entre dos fechas |
| DATENAME | Devuelve una cadena que representa la fecha |
| DATEPART | Devuelve un entero representado una parte de la fecha |
| GETDATE | Obtiene la fecha actual del sistema |

Función DATEADD

DATEADD(*datepart*, *number*, *date*)

- Retorna una nueva fecha después de agregarle *number* expresado como *datepart* a *date*.

```
SELECT  ename, hiredate, DATEADD(mm,5,hiredate)
FROM    emp
WHERE   deptno = 10;
```

| ename | hiredate | |
|--------|-------------------------|-------------------------|
| CLARK | 1981-06-09 00:00:00.000 | 1981-11-09 00:00:00.000 |
| KING | 1981-11-17 00:00:00.000 | 1982-04-17 00:00:00.000 |
| MILLER | 1982-01-23 00:00:00.000 | 1982-06-23 00:00:00.000 |

Función DATEDIFF

DATEDIFF(*datepart*, *startdate*, *enddate*)

- Retorna el número de unidades expresadas como *datepart* entre dos fechas.

```
SELECT  ename, hiredate,  
        DATEDIFF(yy, hiredate, GETDATE()) Years  
FROM    emp  
WHERE   deptno = 20;
```

| ename | hiredate | Years |
|-------|-------------------------|-------|
| SMITH | 1980-12-17 00:00:00.000 | 19 |
| JONES | 1981-04-02 00:00:00.000 | 18 |
| SCOTT | 1982-12-09 00:00:00.000 | 17 |
| ADAMS | 1983-01-12 00:00:00.000 | 16 |
| FORD | 1981-12-03 00:00:00.000 | 18 |

Función DATENAME

- **DATENAME(*datepart*, *date*)**
- **Devuelve una cadena que representa la fecha especificada como *datepart*.**

```
SELECT  ename, hiredate,  
        DATENAME(dw, hiredate) 'Day of week'  
FROM    emp ;
```

| ename | hiredate | Day of week |
|-------|-------------------------|-------------|
| SMITH | 1980-12-17 00:00:00.000 | Wednesday |
| ALLEN | 1981-02-20 00:00:00.000 | Friday |
| WARD | 1981-02-22 00:00:00.000 | Sunday |
| JONES | 1981-04-02 00:00:00.000 | Thursday |
| | | |

Función DATEPART

DATEPART(*datepart*, *date*)

- Devuelve un entero representado una parte de la fecha expresada como *datepart*.

```
SELECT ename, DATEPART(yy, hiredate) Year,  
DATEPART(mm, hiredate) Month, DATEPART(dd,  
hiredate) Day  
FROM emp;
```

| ename | Year |
|-------|------|
| SMITH | 1980 |
| ALLEN | 1981 |
| WARD | 1981 |
| ... | |

Funciones de Conversión



Funciones de Conversión CAST

CAST(expression AS data_type)

➤ **Convierte *expression* al tipo indicado por *data_type*.**

```
SELECT ename, CAST(hiredate AS char) "Hire Date"  
FROM emp;
```

```
SELECT ename ,  
       CAST( hiredate AS char) "Hire Date"  
FROM   emp ;
```

| ename | Hire Date |
|-------|---------------------|
| SMITH | Dec 17 1980 12:00AM |
| ALLEN | Feb 20 1981 12:00AM |
| WARD | Feb 22 1981 12:00AM |
| | |

Funciones de Conversión CONVERT

CONVERT (*data_type*[(*length*)], *expression* [, *style*])

Convierte *expression* al tipo indicado por *data_type* con la longitud dada por *length* utilizando el estilo *style*.

```
SELECT ename, CONVERT(char(10), hiredate, 103) "Hire Date"
FROM emp;
```

| ename | Hire Date |
|-------|------------|
| ----- | ----- |
| SMITH | 17/12/1980 |
| ALLEN | 20/02/1981 |
| WARD | 22/02/1981 |
| | |

Función ISNULL

Convierte un valor nulo a un valor indicado

- **Tipos de datos comunes que se utilizan para evitar un NULL son **date**, **character** y **numeric**.**
- **Ejemplos :**
 - **ISNULL(comm, 0)**
 - **ISNULL(hiredate, GETDATE())**
 - **ISNULL(job, 'No Job Yet')**

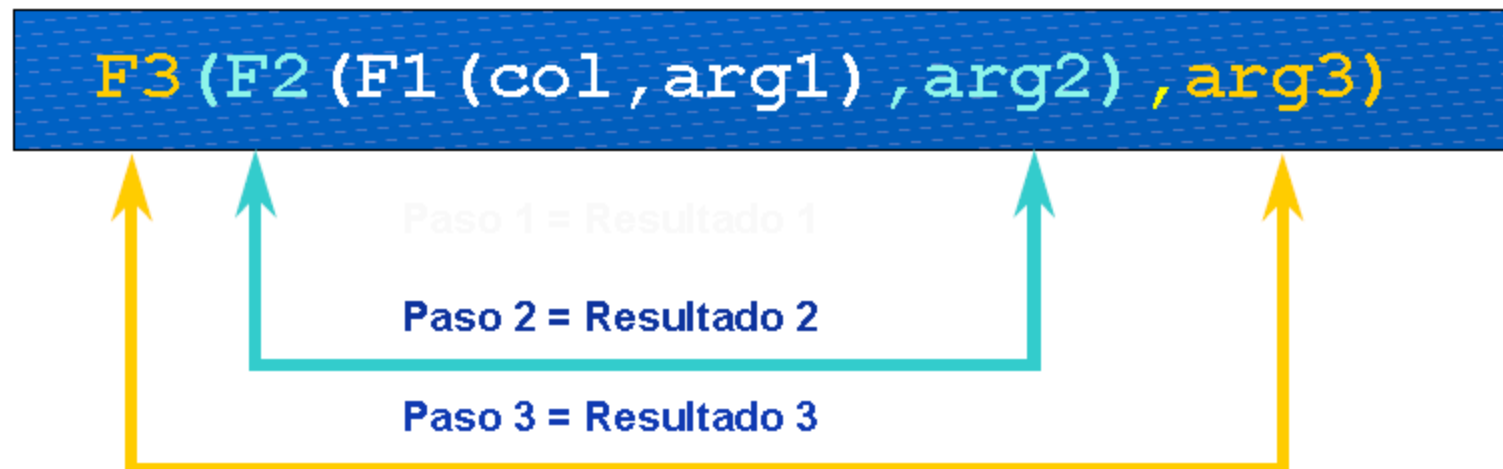
Utilizando la Función ISNULL

```
SELECT  ename, sal, comm, (sal*12)+ISNULL(comm,0)
FROM    emp;
```

| ENAME | SAL | COMM | |
|-------------------|---------|---------|----------|
| SMITH | 800.00 | | 9600.00 |
| ALLEN | 1600.00 | 300.00 | 19500.00 |
| WARD | 1250.00 | 500.00 | 15500.00 |
| JONES | 2975.00 | | 35700.00 |
| MARTIN | 1250.00 | 1400.00 | 16400.00 |
| ... | | | |
| 14 rows selected. | | | |

Funciones Anidadas

- Funciones de Renglón-simple pueden estar anidadas a cualquier nivel.
- Las funciones anidadas son evaluadas desde la más interna a la más externa.



Funciones Anidadas

```
SELECT ename, ISNULL(CONVERT(char(10), mgr), 'No Manager') Manager
FROM emp
ORDER BY mgr;
```

```
SELECT  ename ,
        ISNULL(CONVERT(char(10), mgr), 'No manager') Manager
FROM    emp
ORDER   BY mgr
```

| | |
|-------|------------|
| ename | Manager |
| ----- | ----- |
| KING | No Manager |

Prácticas

- **Crear consultas utilizando funciones numéricas, fecha y carácter**
- **Realizar cálculos sobre fechas**
- **Verificar la fecha de ingreso de los empleados**