Autor: **Nicola Strappazzon**
Web: http://nicola.strappazzon.me
Revisión: 17/11/11

Lección 3

Usando funciones simples para manipular la salida de los registros

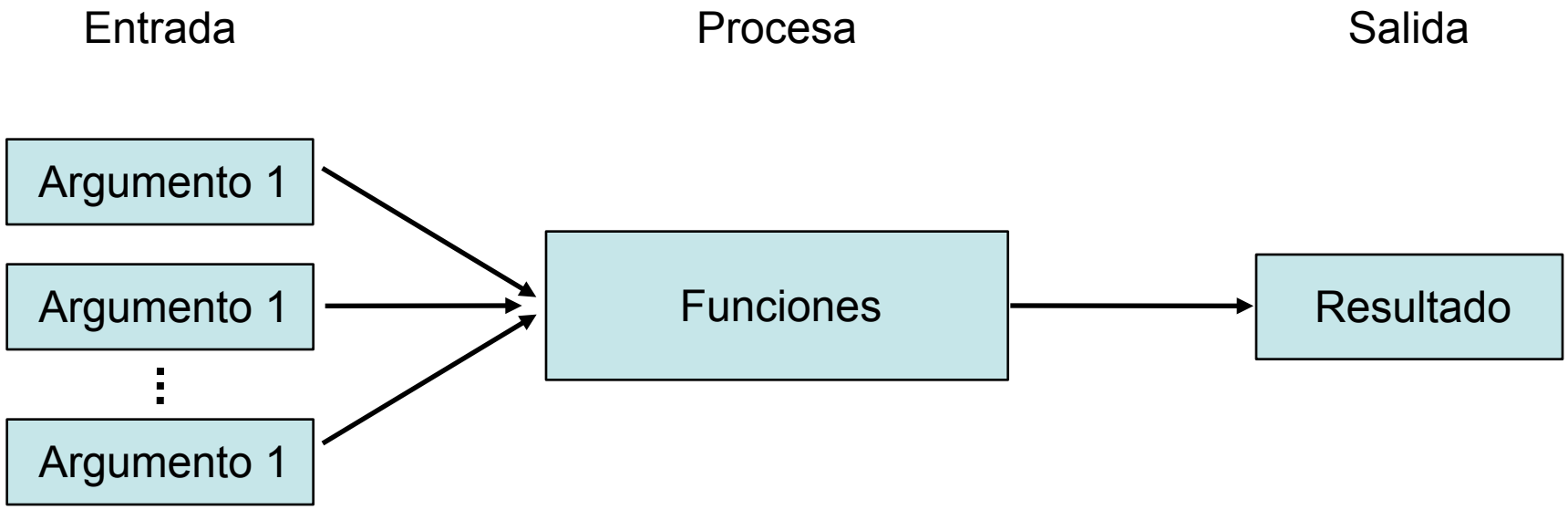
Este material se encuentra basado en el curso de Fundamentos a SQL de ORACLE, el cual es adaptado para el producto PostgreSQL, todos los ejemplos, códigos fuentes y la Base de Datos HR es propiedad de ORACLE.

Objetivos

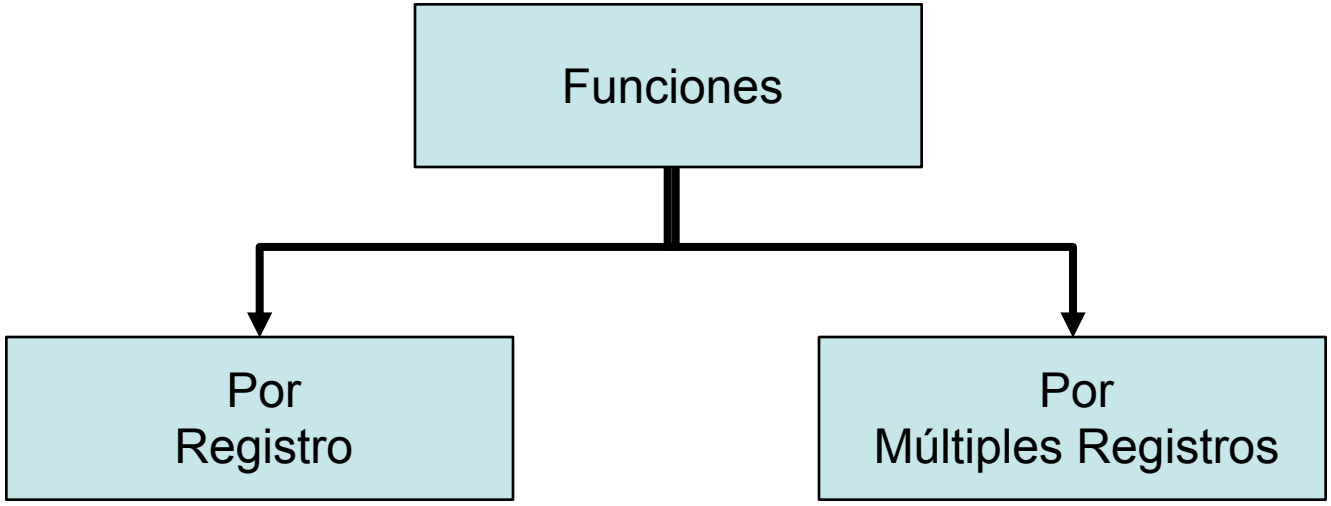
Al completar esta lección usted podrá entender los siguientes puntos:

- Describir diversos tipos de funciones que se encuentran disponibles en SQL.
- Usar funciones que manipulen caracteres, números y fechas dentro de la sentencia **`SELECT`**.
- Usar la función para realizar cambios en los tipos de datos.

Funciones SQL



Dos Tipos de Funciones en SQL

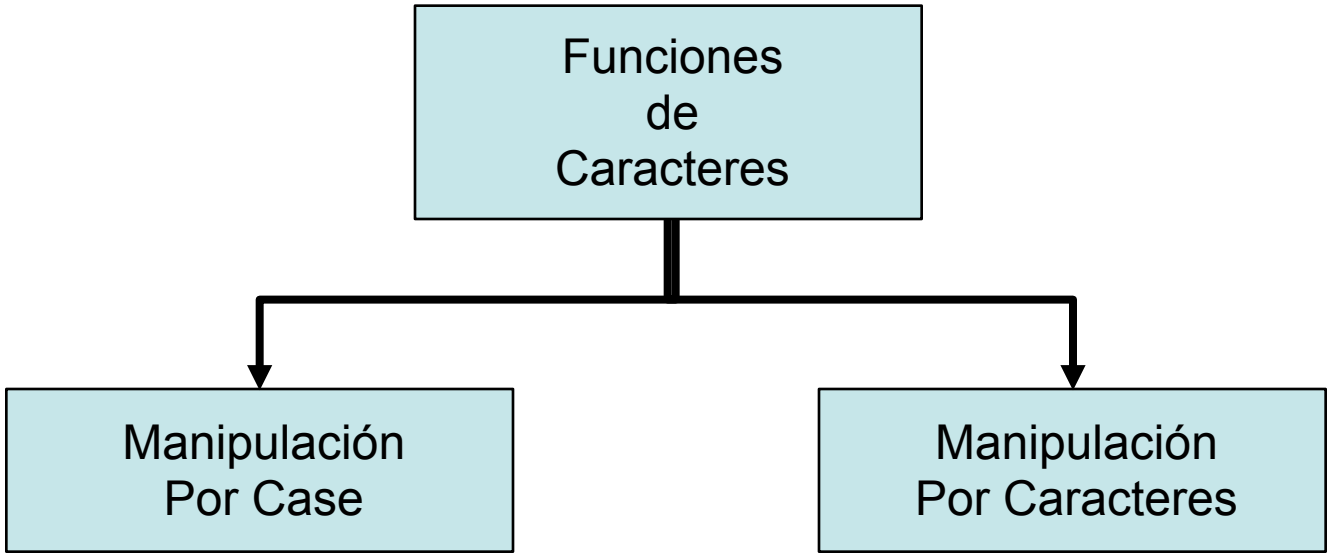


Funciones Por Registro

- Manipula el datos de cada elemento
- Acepta uno o varios argumentos y retorna un solo valor
- Retorna un valor por cada registro
- Puede modificar el tipo de dato
- Puede aceptar como argumento columnas, variables, constantes o expresiones

```
function(arg1, arg2, ...)
```

Funciones de Caracteres



Manipulación del Case

Funciones que modifican la Mayúscula o Minúscula de un texto:

Función	Retorna	Resultado
`lower('TOM')`	`text`	`tom`
`upper('tom')`	`text`	`TOM`
`initcap('hi THOMAS')`	`text`	`Hi Thomas`

Usando la Manipulación del Case

```
SELECT employee_id, last_name, department_id
FROM   employees
WHERE  last_name = 'higgins';
```

```
 employee_id | last_name | department_id
-------------+-----------+---------------
(0 rows)
```

```
SELECT employee_id, last_name, department_id
FROM   employees
WHERE  LOWER(last_name) = 'higgins';
```

```
 employee_id | last_name | department_id
-------------+-----------+---------------
         205 | Higgins   |           110
(1 row)
```

Manipulación de Caracteres

Funciones básicas para la manipulación de texto:

Función	Retorna	Resultado
`substring('Thomas' from 1 for 3)`	text	Tho
`substr('Thomas', 3)`	text	omas
`length('jose')`	int	4
`strpos('high', 'ig')`	int	2
`repeat('Pg', 4)`	text	PgPgPgPg
`rpad('hi', 5, 'xy')`	text	Hixyx
`replace( 'abcdefabcdef', 'cd', 'XX')`	text	abXXefabXXef
`trim(both 'x' from 'xTomxx')`	text	Tom
`md5('abc')`	text	900150983cd24...

Usando la Manipulación de Caracteres

```
SELECT employee_id, LENGTH(last_name),
       SUBSTRING(job_id from 4 for 6)
FROM   employees
WHERE  SUBSTR(job_id, 4) = 'REP';
```

```
 employee_id | length | substring
-------------+--------+-----------
         150 |      6 | REP
         151 |      9 | REP
         152 |      4 | REP
         153 |      5 | REP
         154 |      9 | REP
         155 |      7 | REP
         156 |      4 | REP
         157 |      5 | REP
         158 |      6 | REP
         159 |      5 | REP
(33 rows)
```

Funciones Numéricas

- **`round:`** Redondea un numero decimal.
- **`trunc:`** Trunca la cantidad de decimales a mostrar.
- **`mod:`** Retorna el resto de una división.

Función	Retorna	Resultado
`round(42.4382, 2)`	`int`	`42,44`
`trunc(42.4382, 2)`	`int`	`42,43`
`mod(9,4)`	`int`	`1`

Usando las Función ROUND

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

```
 round | round | round
-------+-------+-------
 45.92 |    46 |    50
(1 row)
```

Usando las Función TRUNC

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

```
 trunc | trunc | trunc
-------+-------+-------
 45.92 |    45 |    40
(1 row)
```

Usando las Función MOD

```
SELECT last_name, salary, MOD(salary, 5000)
FROM   employees
WHERE  job_id = 'SA_REP';
```

```
 last_name   | salary | mod
-------------+--------+------
 Tucker      |  10000 |    0
 Bernstein   |   9500 | 4500
 Hall        |   9000 | 4000
 Olsen       |   8000 | 3000
 Cambrault   |   7500 | 2500
 Tuvault     |   7000 | 2000
 King        |  10000 |    0
 Sully       |   9500 | 4500
 McEwen      |   9000 | 4000
 Smith       |   8000 | 3000
 Doran       |   7500 | 2500
(30 rows)
```

Trabajando con Fechas

- PostgreSQL guarda internamente las fechas bajo el siguiente formato: Siglo, Año, Mes, Día, Horas, Minutos y Segundos.
- Las fechas por defecto se muestran con el formato YYYY-MM-DD, Ejemplo: 2007-12-01.

```
SELECT last_name, hire_date
FROM   employees
WHERE  hire_date < '1988-01-01';
```

```
 last_name | hire_date
-----------+------------
 Whalen    | 1987-09-17
 King      | 1987-06-17
(2 rows)
```

Funciones de Fechas y Horas

Funciones que retornan la Fecha y Hora del sistema:

Función	Resultado
`current_date`	`2007-09-25`
`current_time`	`12:29:59.062-04`
`current_timestamp`	`2007-09-25 12:30:34.218-04`

Operaciones Aritméticas con Fechas

- Sumar o restar una cantidad de días a una fecha.
- Extraer la cantidad de días que han transcurrido entre dos fechas.

Función	Resultado
`date '2001-09-28' + integer '7'`	2001-10-05
`date '2001-09-28' - integer '7'`	2001-09-21
`date '2001-10-01' - date '2001-09-28'`	3

Operaciones Aritméticas con Horas

- Sumar o restar una cantidad de tiempo a una hora.
- Extraer la cantidad de tiempo que a transcurrido entre dos periodos de tiempo.
- Multiplicar una hora por una cantidad de decimales.

Función	Resultado
`time '01:00' + interval '3 hours'`	04:00:00
`time '01:00' + interval '3 minute'`	01:03:00
`time '01:00' + interval '3 seconds'`	01:00:03
`time '04:00' - interval '3 hours'`	01:00:00
`time '04:15' - time '04:00';`	00:15:00
`interval '1 hour' * double precision '3.5'`	03:30:00

Conversiones

- **Implícitas:** Es el tipo de conversión realizada de forma automática por el manejador de base de datos, y no es visible para el usuario.
- **Explicitas:** Cuando la conversión requiere de la intervención del usuario.

NOTA: Existen tipos de datos que no se les pueden realizar una conversión debido a su naturaleza.

Conversión Implícita

```
SELECT '24', integer '24' AS "age";
```

```
 ?column? | age
----------+-----
 24       |  24
(1 row)
```

```
SELECT text 'Origin' AS "label", point '(0,0)' AS "value";
```

```
 label  | value
--------+-------
 Origin | (0,0)
(1 row)
```

Usando la función CAST

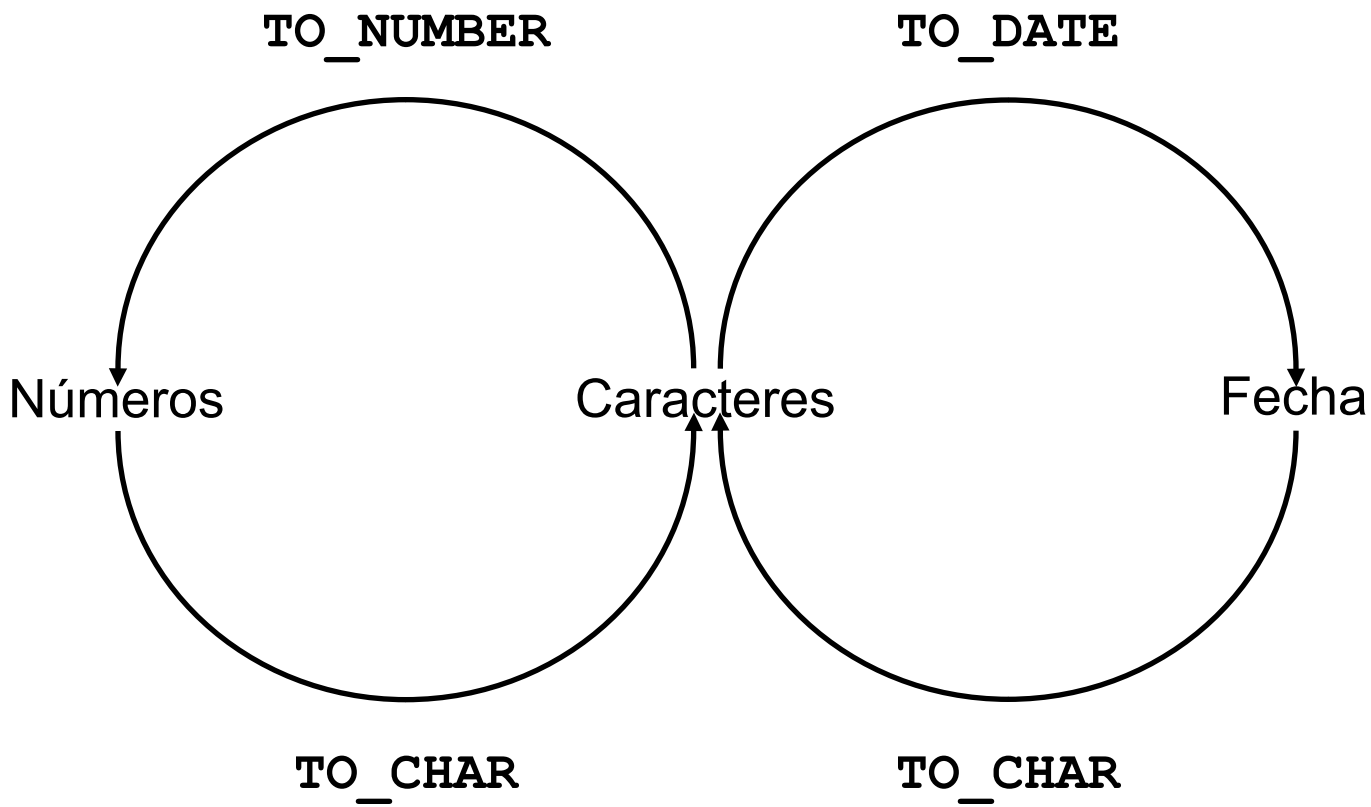
```
SELECT CAST('24' AS INTEGER), CAST('2001-09-10' AS DATE);
```

```
 int4 |    date
------+------------
   24 | 2001-09-10
(1 row)
```

```
SELECT CAST('12.5' AS DECIMAL) - 2;
```

```
 ?column?
----------
     10.5
(1 row)
```

Conversión Explicita



Usando la Función TO_CHAR con Números

```
TO_CHAR(int, 'text')
```

La conversión mediante la función **to_char** permite definir patrones para mostrar el dato bajo el formato deseado.

Elemento	Descripción
9	Valor que representa un dígito.
0	Se muestra el numero 0 en ausencia de un valor.
.	Imprime decimales a partir del punto.
,	Imprime la coma en determinada posición.

Usando la Función TO_CHAR con Números

Función	Resultado
to_char(485, '999')	485
to_char(485, '99')	##
to_char(-485, '999')	-485
to_char(485, '9 9 9')	4 8 5
to_char(1485, '9,999')	1,485
to_char(148.5, '999.999')	148,5
to_char(485, '"Good number:"999')	Good number: 485
to_char(.1, '0.9');	0,1
to_char(1485, '9999.99')	1485

Usando la Función TO_CHAR con Fechas

```
SELECT last_name, TO_CHAR(salary, '$99,999.00') AS "Salary"
FROM employees;
```

```
 last_name  |    Salary
------------+--------------
 Russell    | $ 14,000.00
 Partners   | $ 13,500.00
 Errazuriz  | $ 12,000.00
 Cambrault  | $ 11,000.00
 Zlotkey    | $ 10,500.00
 Tucker     | $ 10,000.00
 Bernstein  | $  9,500.00
 Hall       | $  9,000.00
 Olsen      | $  8,000.00
 Cambrault  | $  7,500.00
 Tuvault    | $  7,000.00
 King       | $ 10,000.00
(107 rows)
```

Usando la Función TO_CHAR con Fechas

```
TO_CHAR(date, 'text')
```

La conversión mediante la función **to_char** permite definir patrones para mostrar el dato bajo el formato deseado.

Elemento	Descripción
YYYY	Año de cuatro dígitos.
YY	Año de dos dígitos.
MONTH	Nombre del mes en mayúscula.
MM	Numero del mes (01-12).
DAY	Nombre del día en mayúscula.
DY	Nombre abreviado del día en 3 letras.
DD	Numero del día (01-31).

Usando la Función TO_CHAR con Fechas

```
SELECT last_name,
       TO_CHAR(hire_date, 'YYYY, MONTH') AS "Hire Date"
FROM employees;
```

```
 last_name  |     Hire Date
------------+------------------
 Russell    | 1996, OCTOBER
 Partners   | 1997, JANUARY
 Errazuriz  | 1997, MARCH
 Cambrault  | 1999, OCTOBER
 Zlotkey    | 2000, JANUARY
 Tucker     | 1997, JANUARY
 Bernstein  | 1997, MARCH
 Hall       | 1997, AUGUST
 Olsen      | 1998, MARCH
 Cambrault  | 1998, DECEMBER
 Tuvault    | 1999, NOVEMBER
(107 rows)
```

Patrones para modificar el formato

Utilizar el modificador TM (Translation Mode) para imprimir los nombres de los días y meses según la configuración del Sistema Operativo, se puede utilizar en cualquier patrón.

```
SELECT to_char(current_timestamp, 'TMDay, TMMonth-yyyy');
```

```
      to_char
-------------------
 Lunes, Enero-2010
(1 row)
```

Usando las funciones TO_NUMBER y TO_DATE

- Convierte una cadena de caracteres a un formato numérico usando la función **TO_NUMBER**:

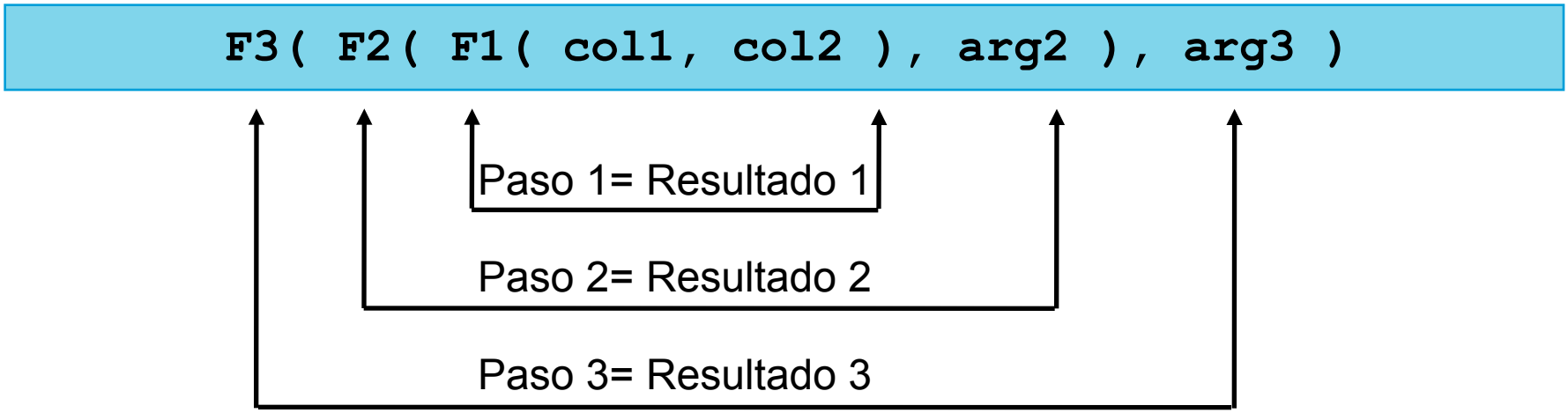
```
TO_NUMBER(text[, 'text'])
```

- Convierte una cadena de caracteres a una fecha con formato usando la función **TO_DATE**:

```
TO_DATE(text[, 'text'])
```

Funciones Jerárquicas

- Dentro de un registro pueden haber distintas funciones en muchos niveles.
- Las funciones jerárquicas son evaluadas de adentro hacia a fuera.



Funciones Jerárquicas

```
SELECT last_name,
       UPPER(SUBSTR(last_name, 1, 8) || '_US')
FROM   employees
WHERE  department_id = 60;
```

```
 last_name |    upper
-----------+-------------
 Hunold    | HUNOLD_US
 Ernst     | ERNST_US
 Austin    | AUSTIN_US
 Pataballa | PATABALL_US
 Lorentz   | LORENTZ_US
(5 rows)
```

Expresiones Condicionales

Las siguientes funciones trabajan con cualquier tipo de dato y permite el uso del valor NULL:

- **`COALESCE`**
- **`CASE`**

Función COALESCE

- Convierte NULL a un valor en especifico.
- El valor devuelto debe ser del mismo tipo al que se encuentra definido en la columna.
- Soporta los tipos de datos numéricos, caracteres y fechas.

```
COALESCE(value, [, ...]);
```

Función COALESCE

```
SELECT last_name,
       COALESCE(manager_id, commission_pct, -1) AS comm
FROM   employees
ORDER BY commission_pct;
```

```
  last_name   | comm
--------------+------
 Marvins      |  147
 Johnson      |  149
 Baer         |  101
 Higgins      |  101
 Gietz        |  205
 King         |   -1
 Jones        |  123
 Kochhar      |  100
 De Haan      |  100
 Hunold       |  102
 (107 rows)
```

Función COALESCE

```
COALESCE(value1, value2, value3, ... valueN);
```

- **value1:** Retorna este valor si no es NULL.
- **value2:** Retorna este valor si no es NULL y si el primer valor (value1) es NULL.
- **value3:** Retorna este valor si todas las anteriores son NULL.

Función COALESCE

```
SELECT last_name,
       COALESCE(commission_pct, 0) AS comm
FROM   employees
ORDER BY commission_pct;
```

```
  last_name   | comm
--------------+------
 Marvins      | 0.10
 Johnson      | 0.10
 Kumar        | 0.10
 Lee          | 0.10
 King         | 0.35
 McEwen       | 0.35
 Russell      | 0.40
 Gee          |    0
 Philtanker   |    0
 Ladwig       |    0
(107 rows)
```

Función CASE

- Facilita el uso de la condición **IF-THEN-ELSE** mediante la sintaxis de SQL.

```
CASE WHEN condition THEN result
     [WHEN ...]
     [ELSE result]
END;
```

Función CASE

```
SELECT last_name, job_id, salary,
       CASE job_id WHEN 'SA_MAN' THEN 1.10 * salary
                   WHEN 'SA_REP' THEN 1.15 * salary
                   ELSE salary
       END AS "New Salary"
FROM   employees;
```

```
 last_name  |   job_id   | salary | New Salary
------------+------------+--------+------------
 Russell    | SA_MAN     |  14000 |   15400.00
 Partners   | SA_MAN     |  13500 |   14850.00
 Errazuriz  | SA_MAN     |  12000 |   13200.00
 Cambrault  | SA_MAN     |  11000 |   12100.00
 Zlotkey    | SA_MAN     |  10500 |   11550.00
 Tucker     | SA_REP     |  10000 |   11500.00
 Bernstein  | SA_REP     |   9500 |   10925.00
 Hall       | SA_REP     |   9000 |   10350.00
(107 rows)
```

Resumen

En esta lección, usted debió entender como:

- Usar las funciones básicas de SQL.
  - Manipular las cadenas de caracteres.
  - Modificar decimales.
  - Manipular fechas.
  - Realizar conversiones entre los principales Tipos de Datos.
  - Expresiones de Condición.