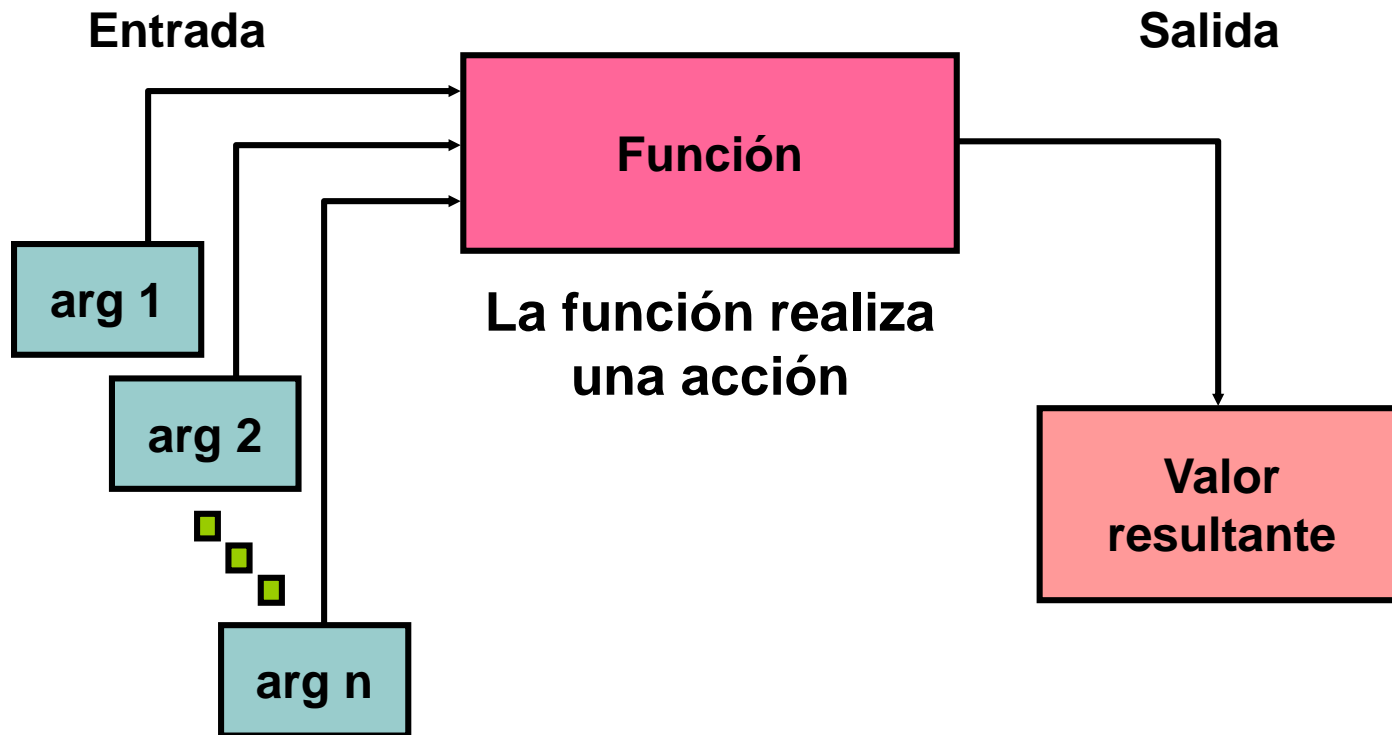


FUNCIONES DE UNA SOLA FILA

Funciones SQL



Funciones

- Existen funciones poderosas incorporadas en SQL que pueden ser usadas para:
 - Realizar cálculos con los datos
 - Modificar ítems de datos individuales
 - Manipular la salida de grupos de registros
 - Alterar los formatos de fecha y números para su despliegue
 - Convertir tipos de datos de las columnas

Tipos de funciones SQL

- Funciones de una sola fila

- Character
- Number
- Date
- Conversion



- Funciones de múltiples filas

- Group



Funciones de una Fila

Sintaxis: *Nombre_de_función (columna | expresión, [arg1, arg2,...])*

- Son utilizadas para manipular ítems de datos
- Pueden aceptar uno o más argumentos pero siempre retornan un solo valor
- Actúan sobre cada fila retornada devolviendo un resultado por fila
- Pueden modificar el tipo de datos
- Las funciones pueden usarse de forma anidada

LA TABLA DUAL

- Es una tabla que se crea automáticamente junto con el diccionario de datos de Oracle. Solo posee una columna: "Dummy", definida como de tipo Varchar2(1), y contiene una única fila, con el valor X
- Es útil para realizar cálculos, evaluar expresiones o retornar constantes mediante la sentencia `SELECT`

Funciones para Strings

Función	Propósito
LOWER (columna expresión)	Convierte un string alfanumérico a minúsculas
UPPER (columna expresión)	Convierte un string alfanumérico a mayúsculas
INITCAP (columna expresión)	Convierte un string alfanumérico transformando la letra inicial de cada palabra a mayúsculas y todas las otras en minúsculas
CONCAT (columna expresión , columna2 expresión2)	Equivalente al operador de concatenación
SUBSTR (columna expresión, m [,n])	Retorna una cantidad de caracteres de la expresión, partiendo de la posición m, con n caracteres de longitud. Si m es negativo, el conteo inicia desde el final.
LENGTH (columna expresión)	Retorna el número de caracteres
NVL (columna expresión1, columna expresión1)	Si expresión1 es NULL, convierte al valor de la segunda expresión.

Funciones para Strings

Función	Propósito
INSTR (columna expresión, 'carácter' [,m] [,n])	Retorna la posición numérica del 'carácter'. Opcionalmente, se puede indicar que la búsqueda inicie de la posición m, y también se puede indicar que busque la ocurrencia n del carácter. m y n por defecto tienen valor 1. Si m es negativo, la búsqueda inicia desde el final
LTRIM (columna expresión [, 'caracteres'])	Quita todas las apariciones del conjunto de 'caracteres' del lado izquierdo de la columna o expresión. Ver la función TRIM.
RTRIM (columna expresión [, 'caracteres'])	Quita todas las apariciones del conjunto de 'caracteres' del lado derecho de la columna o expresión. Ver la función TRIM.
TRIM (LEADING TRAILING BOTH 'caracteres' FROM columna expresión)	Permite quitar los caracteres de adelante (leading), atrás (trailing) o de ambos lados de la columna o expresión. Por defecto, el TRIM actúa de ambos lados (BOTH).
LPAD (string, n, ['caracteres'])	Justifica el string a la derecha, llenando el espacio con los caracteres indicados. El string total resultante tiene un total de 'n' de longitud
RPAD (string, n, ['caracteres'])	Justifica el string a la izquierda, llenando el espacio con los caracteres indicados. El string total resultante tiene un total de 'n' de longitud

Funciones para Strings

Función	Resultado
<code>LOWER('SQL Course')</code>	<code>sql course</code>
<code>UPPER('SQL Course')</code>	<code>SQL COURSE</code>
<code>INITCAP('SQL Course')</code>	<code>Sql course</code>

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

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

Funciones para Strings

Función	Resultado
<code>CONCAT('Hello', 'World')</code>	<code>HelloWorld</code>
<code>SUBSTR('HelloWorld',1,5)</code>	<code>Hello</code>
<code>LENGTH('HelloWorld')</code>	<code>10</code>
<code>INSTR('HelloWorld', 'W')</code>	<code>6</code>
<code>LPAD(salary,10,'*')</code>	<code>*****24000</code>
<code>RPAD(salary, 10, '*')</code>	<code>24000*****</code>
<code>REPLACE('JACK and JUE','J','BL')</code>	<code>BLACK and BLUE</code>
<code>TRIM('H' FROM 'HelloWorld')</code>	<code>elloWorld</code>

Funciones numéricas

Función	Propósito
ROUND (<i>columna expresión, n</i>)	Redondea la columna o expresión a n lugares decimales. Si se omite n , se suprimen las posiciones decimales. Si n es negativo, se redondean los dígitos a la izquierda del punto decimal
TRUNC (<i>columna expresión, n</i>)	Trunca la columna o valor a n posiciones decimales, y si se omite n , no existen posiciones decimales. Si n es negativo, los números a la izquierda del punto decimal son truncados a cero.
MOD (<i>m, n</i>)	Retorna el resto de m dividido n

Funciones Numéricas

- ROUND: Redondea el valor al decimal especificado
- TRUNC: Trunca el valor al decimal especificado
- MOD: Devuelve el resto de la división

Función	Resultado
ROUND (45.926, 2)	45.93
TRUNC (45.926, 2)	45.92
MOD (1600, 300)	100

Formato de Fechas

- ORACLE almacena fechas internamente en formato numérico: Centurias, año, meses, días , horas, minutos y segundos
- Por default se despliega en el formato DD-MM-YYYY
- SYSDATE es una función que retorna fecha y hora.
- SYSDATE no tiene argumentos

```
SELECT sysdate FROM dual;
```

Operaciones con Fechas

Operación	Resultado o	Descripción
Fecha + número	Fecha	Suma un número de días a una fecha
Fecha – número	Fecha	Resta un número de días a una fecha
Fecha – fecha	Número de días	Resta una fecha de otra
Fecha + número/24	Fecha	Suma un número de horas a una fecha

Funciones de Fecha

Función	Propósito
MONTHS_BETWEEN (fecha1, fecha2)	Devuelve el número de meses entre fecha1 y fecha2. Si fecha1 < fecha2 entonces devuelve negativo
ADD_MONTHS (fecha, n)	Suma n meses a la fecha (n entero y positivo)
NEXT_DAY (fecha, 'día de la semana')	Encuentra la fecha que corresponde al siguiente día de la semana. Este 'día de la semana' puede expresarse como literal o como número
LAST_DAY (fecha)	Devuelve el último día del mes correspondiente a la 'fecha' especificada
ROUND (fecha [, formato])	Encuentra la fecha correspondiente al primer del mes contenido en 'fecha' si se especifica un formato 'MM', si se especifica por ejemplo 'YEAR', devuelve el primer día del año.
TRUNC (fecha [, formato])	Retorna la misma fecha, seteada a la medianoche. Es útil para remover la porción de horario de los campos tipo date.

Funciones de Fecha

Función	Resultado
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-FEB-95')	'28-FEB-95'

- Supongamos que SYSDATE = '25-JUL-03':

Función	Resultado
ROUND(SYSDATE,'MONTH')	01-AUG-03
ROUND(SYSDATE ,'YEAR')	01-JAN-04
TRUNC(SYSDATE ,'MONTH')	01-JUL-03
TRUNC(SYSDATE ,'YEAR')	01-JAN-03

Obtener una parte de la fecha con EXTRACT

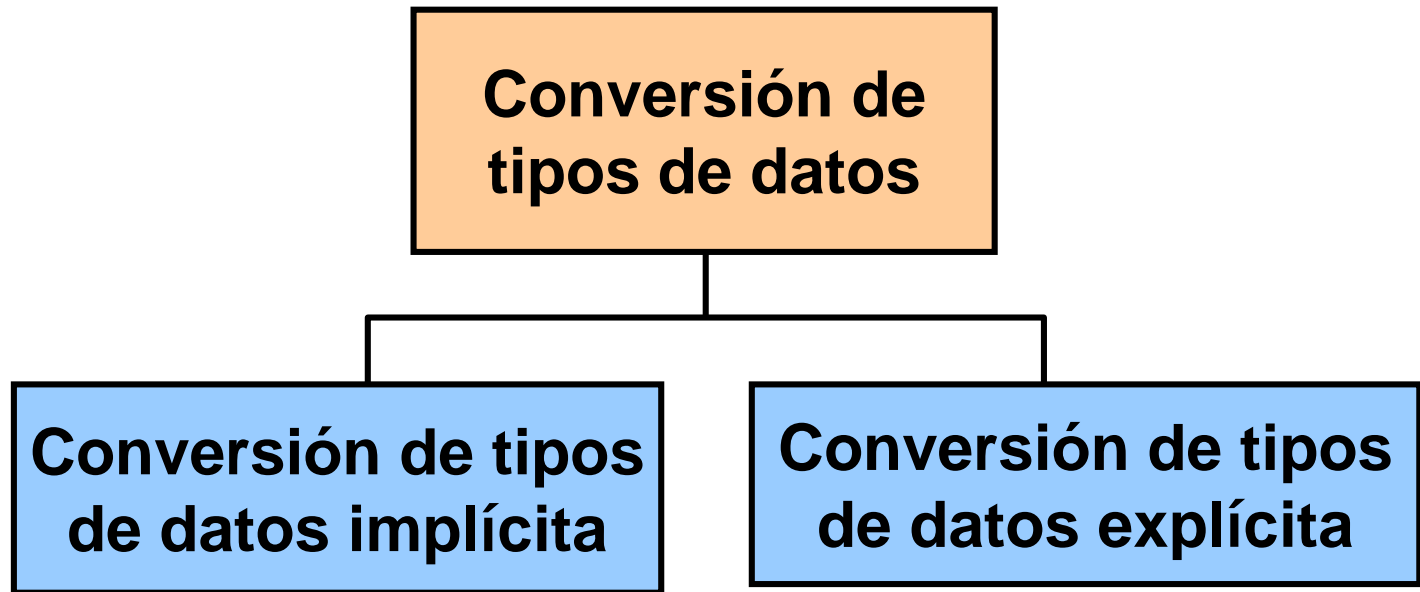
- EXTRACT es una función de fecha que permite obtener una parte de la fecha de los campos DATE o TIMESTAMP. Ejemplos:

```
SELECT EXTRACT (YEAR FROM SYSDATE) FROM DUAL;  
SELECT EXTRACT (MONTH FROM SYSDATE) FROM DUAL;
```

EXTRACT puede utilizarse para extraer:

- AÑO (YEAR)
- MES (MONTH)
- DÍA (DAY)
- HORA (HOUR)
- MINUTO (MINUTE)
- SEGUNDO (SECOND)

Funciones de Conversión



Conversión de Tipos de Datos Implícita

- Para las asignaciones, Oracle Server puede convertir automáticamente:

De	A
VARCHAR2 o CHAR	NUMBER
VARCHAR2 o CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

Funciones de conversión

Función	Propósito
TO_CHAR(numero fecha, ['formato'])	Convierte un número o fecha a VARCHAR2 con el formato dado.
TO_NUMBER('char')	Convierte a números un string que contiene dígitos numéricos
TO_DATE('char', ['formato'])	Convierte un string que representa una fecha, a date, de acuerdo al formato especificado. Si se omite el formato, por defecto asume 'DD-MON-YY'.

Uso de la Función TO_CHAR con Fechas

```
TO_CHAR(date, 'format_model') 
```

- El modelo de formato:
 - Debe ir entre comillas simples
 - Es sensible a mayúsculas/minúsculas
 - Puede incluir cualquier elemento de formato de fecha válido
 - Tiene un elemento fm para eliminar espacios en blanco de relleno o suprimir ceros iniciales
 - Está separado del valor de fecha por una coma

Uso de la Función TO_CHAR con Fechas

```
SELECT last_name,  
       TO_CHAR(hire_date, 'fmDD Month YYYY') AS HIREDATE  
FROM   employees;
```

LAST_NAME	HIREDATE
King	17 June 1987
Kochhar	21 September 1989
De Haan	13 January 1993
Hunold	3 January 1990
Ernst	21 May 1991
Lorentz	7 February 1999
Mourgos	16 November 1999

Formatos de fecha

- YYYY representa los 4 dígitos del año
- YEAR representa el año expresado en letras (en Inglés)
- MM representa los 2 dígitos del mes
- MONTH representa el mes expresado en letras
- DY representa las 3 primeras letras de los días de semana
- DAY representa un día de la semana
- DD son 2 dígitos representando el día.
- HH, HH12 ó HH24 representa una hora (1-12) o (0 a 23)
- MI el minuto (0-59)
- SS segundos (0-59)
- RR igual que YY, pero interpreta de acuerdo al siglo actual

Elementos del Modelo de Formato de Fecha

- Los elementos horarios formatean la parte de hora de la fecha:

<code>HH^o24:MI:SS AM</code>	15:45:32 PM
---	--------------------

- Agregue cadenas de caracteres poniéndolas entre comillas dobles:

<code>DD "of" MONTH</code>	12 of OCTOBER
-----------------------------------	----------------------

- Utilice sufijos para escribir el nombre completo de los números:

<code>ddspth</code>	fourteenth
----------------------------	-------------------

Uso de la Función TO_CHAR con Números

TO_CHAR(*number*, '*format_model*') ddspth

- Éstos son algunos de los elementos de formato que se pueden utilizar con la función TO_CHAR para mostrar un valor numérico como carácter:

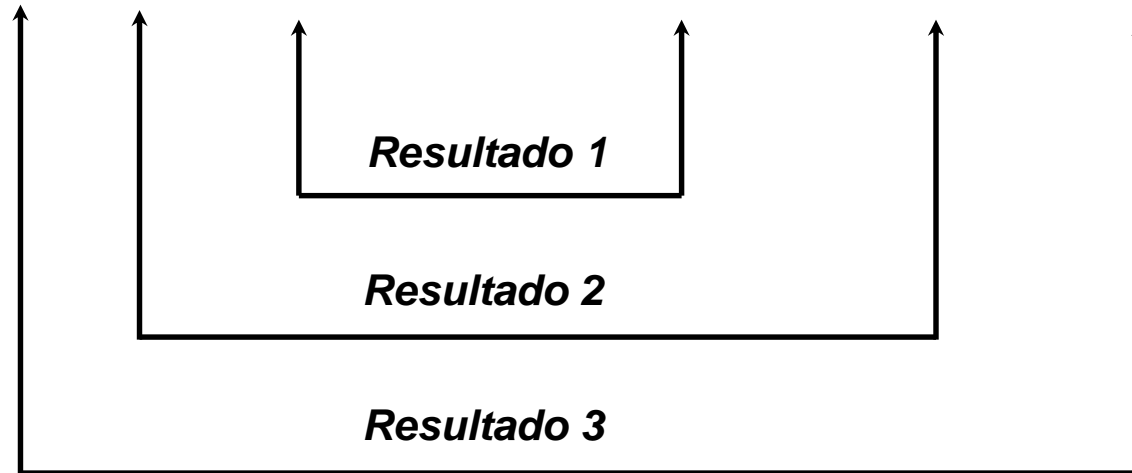
Elemento	Resultado
9	Representa un número
0	Muestra ceros
\$	Coloca un signo de dólar flotante
L	Utiliza el símbolo de divisa local flotante
.	Imprime un punto decimal
,	Imprime una coma como indicador de miles

Formatos de números

Elemento	Descripción	Ejemplo	Resultado
9	Posición numérica (el número de 9s determina la longitud del campo desplegado)	99999	1234
0	Despliega el número precedido de 0s	099999	001234
\$	Signo flotante que representa la moneda (dólar)	\$99999	\$1234
D	Representa el carácter de la posición decimal (, ó .) según la configuración local. Por defecto es el .	99D99	99,99
G	Representa el carácter de la posición de miles (, ó .) según la configuración local	999G999	999.999

Funciones Anidadas

F3 (F2 (F1(columna, arg1), arg2), arg3)



Anidamiento de Funciones

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

LAST_NAME	UPPER(CONCAT(SUBSTR(LAST_NAME,1,8
Hunold	HUNOLD_US
Ernst	ERNST_US
Lorentz	LORENTZ_US

Funciones Generales

Función	Descripción
NVL (expr1,expr2)	Si expresión1 es nula, convierte a expresión2
NVL2 (expr1,expr2, expr3)	Si expr1 no es nulo, devuelve expr2. Si expr1 es nulo, devuelve expr3. El argumento expr1 puede tener cualquier tipo de datos.
NULLIF (expr1,expr2)	Compara 2 expresiones y retorna nulo si ambas son iguales. Retorna la primera expresión si no son iguales
COALESCE expr1, expr2, .., expresiónN)	Retorna la primera expresión no nula en una lista de expresiones

NVL

- Convierte NULL al valor indicado por la expresión NVL
- Puede ser utilizado en tipos date, number, varchar2
- El tipo de datos de la columna a convertir debe ser equivalente al asignado por NVL.
 - NVL (fecha_ini, '01-JAN-95')
 - NVL (RUC, 'No tiene RUC')
 - NVL (asignacion, 1000)

```
SELECT CEDULA, NVL(RUC, ' (s/ RUC) '),  
NOMBRE, APELLIDO FROM B_PERSONAS;
```

La función COALESCE

- La ventaja del COALESCE sobre la función NVL, es que a diferencia de éste, el COALESCE puede tomar múltiples alternativas
- Si la primera expresión es no nula, el COALESCE retorna esa expresión, en caso contrario retorna la segunda. Si ésta a su vez es nula, retorna la siguiente y así sucesivamente

```
SELECT COALESCE(RUC,CEDULA) IDENTIFICACION,  
NOMBRE, APELLIDO FROM B_PERSONAS;
```

Expresiones Condicionales

- Permiten utilizar la lógica IF-THEN-ELSE dentro de una sentencia SQL
- Puede usar dos métodos:
 - Expresión CASE
 - Función DECODE

La expresión CASE

- Una funcionalidad parecida al DECODE es la que ofrece la expresión CASE:

```
CASE <expresion>
```

```
    WHEN <valor1>      THEN <resultado1>
```

```
    WHEN <valor2>      THEN <resultado2>
```

```
    WHEN <valor3>      THEN <resultado3>
```

```
    [ELSE <resultado>]
```

```
END
```

La función DECODE

DECODE (valor, if1, then1 [,if2, then2,]....,else)

- Valor: representa una variable o columna a ser evaluada, cuyo valor se 'testea' o prueba.
- Si if1 es verdadero, entonces el resultado del DECODE es then1

La función DECODE (2)

- Si if2, entonces el resultado es then2, y así sucesivamente,
- El valor else, es obligatorio

```
SELECT nombre, apellido, DECODE(tipo_persona,  
    'F','Fisica', 'J', 'Jurídica', 'Tipo no  
    definido') "tipo persona"  
FROM B_PERSONAS;
```

La expresión CASE: Ejemplo

```
SELECT nombre, apellido,  
       CASE tipo_persona  
         WHEN 'F' THEN 'Física'  
         WHEN 'J' THEN 'Jurídica'  
         ELSE 'Tipo no definido'  
       END "tipo persona"  
FROM B_PERSONAS;
```