# TRABAJO ASINCRÓNICO SEMANA 9

# Unidad 2 - Funciones de una sola fila en SQL

NOMBRE: Bárbara Carvajal Sáez

CARRERA: Ingeniería en Informática

ASIGNATURA: Base de Datos Relacionales

PROFESOR: Francisco Prieto Rossi

FECHA: octubre 2022

# Introducción

Para el siguiente trabajo en grupo, ustedes deberán tener creada una cuenta en Oracle Cloud (*Referencias de Creación en el Ambiente de Aprendizaje Inacap*), usando las estructuras desarrolladas en la primera actividad asincrónica de la asignatura

Por lo anterior definido, se tiene los siguientes tres usuarios:

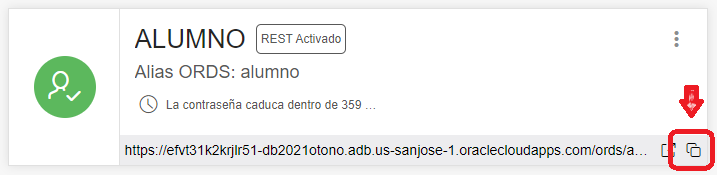
• Gobierno

• Inacap

• Alumno

La contraseña de todos ellos debe ser “oracle4U\_2021” (no debe ser cambiada).

Escriban a continuación, el enlace para acceder a la cuenta alumno en su máquina de trabajo Cloud



**Alumno nota: no me deja poner bien el link, porfavor ver al pegar el link quitar el espacio que hay antes del 1, mil disculpas. O hacer click donde dice link.**

[**Link:**](https://g82171fa0568443-pruebaasincronica.adb.sa-santiago-1.oraclecloudapps.com/ords/alumno/_sdw/) **https://g82171fa0568443-pruebaasincronica.adb.sa-santiago-1.oraclecloudapps.com/ords/alumno/\_sdw/**

Existen ejercicios abiertos de desarrollo por el grupo de trabajo. El puntaje va en función a lo complejo del desarrollo y la argumentación dada. En caso de existir ejercicios de este tipo, iguales en dos o más grupos distintos, se anula la respuesta

El presente trabajo se realiza en la clase y debe ser entregado el mismo día de clases (con una extensión hasta el sábado a las 12:00)

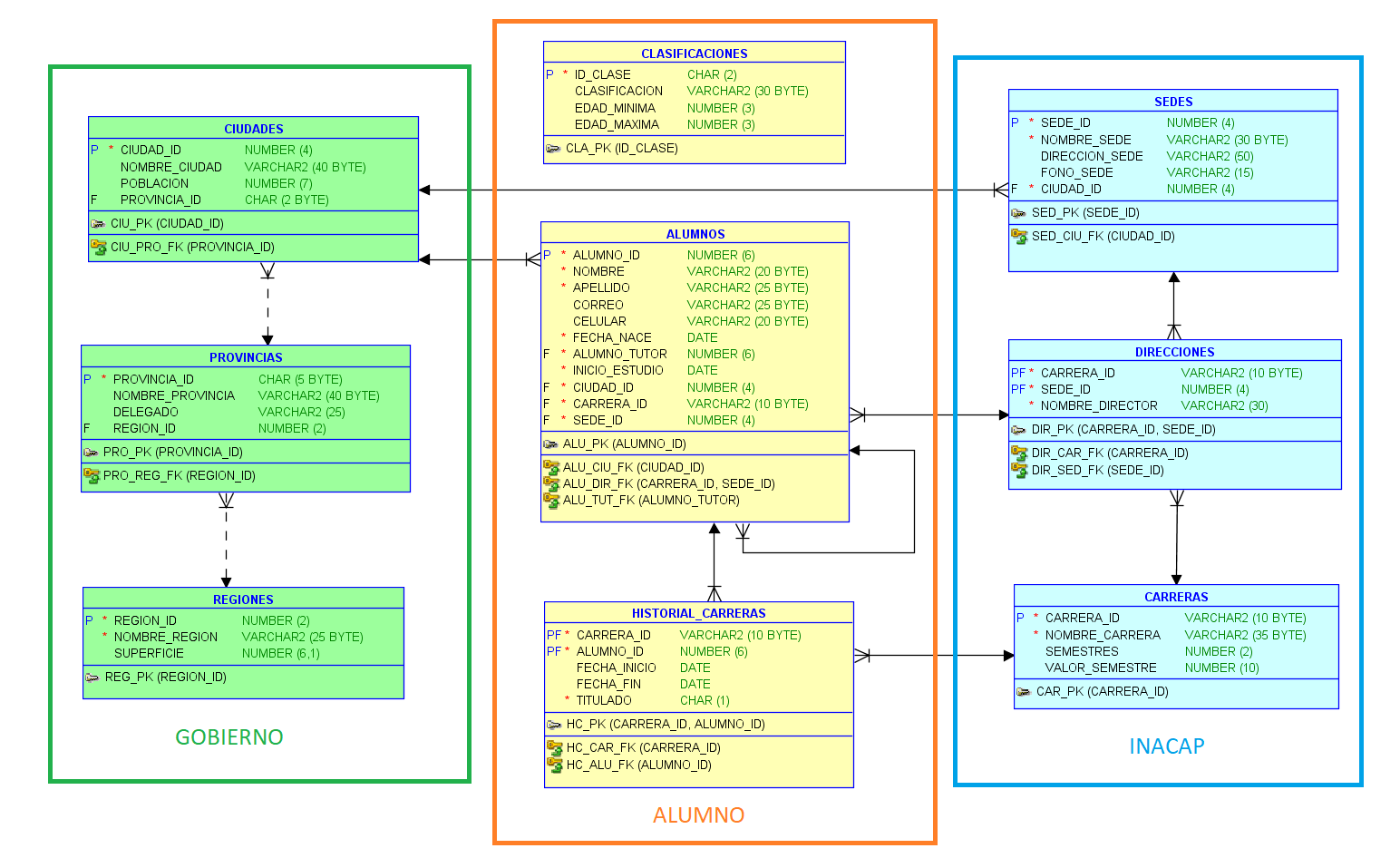
(\*) IMPORTANTE: para cada respuesta dada, deben indicar quien es el integrante del grupo que generó la respuesta, en caso de requerir consulta por parte del docente (debe estar equilibrado entre los integrantes)

# Modelo de Datos

Para este modelo de datos, se consideran tablas compartidas en las distintas cuentas de Gobierno, Inacap y Alumno

Deben respetarse los tipos de datos de cada tabla, y verificar que se tengan los accesos correspondientes según lo expuesto en la actividad anterior, trabajada en grupo

Los datos a considerar, son los originales que se encuentran en los scripts proporcionados en la actividad anterior. Cualquier cambio de dato, debe ser documentado y argumentado



# Desarrollo de consultas a la base de datos

Escriba el comando SQL que es requerido para la consulta planteada (*primero explicando que se va a mostrar en la consulta, y luego el código SQL correspondiente*):

1. Uso de las Funciones de Grupo.  
   Uniendo al menos dos tablas del modelo para cada consulta y usando un nivel de agrupamiento (GROUP BY), desarrolle **dos casos** por cada una de las siguientes funciones grupales:
   1. Sum()

-- sum 1 sumar la población total según provincia

SELECT PROVINCIA\_ID ID,

SUM(POBLACION) POBLACION

FROM gobierno.ciudades

INNER JOIN GOBIERNO.PROVINCIAS USING(PROVINCIA\_ID)

GROUP BY PROVINCIA\_ID

ORDER BY POBLACION DESC;

-- sum 2 sumar el valor total de las carreras por semestre agrupados por el nombre de la sede.

select nombre\_sede, sum(valor\_semestre) "VALOR TOTAL CARRERAS POR SEMESTRE"

from INACAP.sedes, INACAP.CARRERAS, INACAP.DIRECCIONES

where sedes.sede\_id = direcciones.sede\_id

GROUP by nombre\_sede;

* 1. Count()

-- count 1 contar el total de provincias según región.

SELECT NOMBRE\_REGION, count(PROVINCIA\_ID) "TOTAL PROVINCIAS"

from gobierno.REGIONES

LEFT JOIN gobierno.PROVINCIAS

USING(REGION\_ID)

group by NOMBRE\_REGION

ORDER by "TOTAL PROVINCIAS";

* 1. Max()

-- 1.3. Max() MUESTRA LA PROVINCIA CON MÁS ALTA POBLACIÓN

SELECT NOMBRE\_PROVINCIA, MAX(POBLACION ) "POBLACION MÁS ALTA"

FROM GOBIERNO.PROVINCIAS

INNER JOIN GOBIERNO.ciudades

USING(PROVINCIA\_ID)

GROUP BY "NOMBRE\_PROVINCIA";

* 1. Min()

-- min muestra la población menos poblada y agregándoles un

-- filtro de que muestre la ciudad que tenga menos de 5000 pobladores

SELECT

NOMBRE\_CIUDAD,

MIN(POBLACION ) "CIUDAD MENOS POBLADA"

FROM GOBIERNO.CIUDADES

INNER JOIN GOBIERNO.PROVINCIAS

USING(PROVINCIA\_ID)

WHERE POBLACION<5000

GROUP BY NOMBRE\_CIUDAD

ORDER BY NOMBRE\_CIUDAD;

* 1. Avg()

-- MOSTRAR EL PROMEDIO DE LA POBLACION

SELECT NOMBRE\_PROVINCIA PROVINCIA,

ROUND(AVG( POBLACION ),2)"PROMEDIO POBLACION"

FROM GOBIERNO.PROVINCIAS

INNER JOIN GOBIERNO.CIUDADES

USING(PROVINCIA\_ID)

GROUP BY NOMBRE\_PROVINCIA;

1. Uso de funciones de Carácter
   1. Realice una consulta en la que utilice Lower, Upper e Initcap (una de estas funciones debe ser usada en la condición Where)

SELECT concat(concat(upper(NOMBRE\_CIUDAD),

initcap(': hay ')),

concat(POBLACION,

lower(' habitantes'))) CENSO\_COQUIMBO

FROM gobierno.CIUDADES

WHERE NOMBRE\_CIUDAD = 'Coquimbo';

* 1. Usando las tablas en el esquema Gobierno, realice 3 consultas usando funciones de manipulación[[1]](#footnote-1) de caracteres (cada consulta utilizando dos funciones distintas)

--Mostrar última letra de nombre de la región, además en minúscula

SELECT nombre\_region ,

lower(substr (nombre\_region,-1))

from gobierno.regiones;



-- mostrar un "\_" cuando haya un espacio entre los

-- nombres de las ciudades además de transformar en mayúscula.

SELECT nombre\_ciudad ,

replace(upper(nombre\_ciudad),' ','\_')

from gobierno.ciudades;



--Mostrar todas las provincias con un largo menor a 8 y rellenar

--a la izquierda con “#” hasta que hayan en total 13 caracteres.

SELECT nombre\_provincia,

lpad(nombre\_provincia,13,'#')

from gobierno.provincias

where length (nombre\_provincia)<8;

* 1. Usando una de las tablas en el esquema Alumno, realizar una consulta que anide tres funciones de manipulación1 de caracteres

-- crear correo electrónico para los alumnos de inacap junto su nombre + sede y concatenando @alumnoinacap.cl

-- además donde haya una letra "A" agregarle un A.

select nombre ALUMNO, nombre\_sede SEDE,

Replace(CONCAT(upper(concat(substr(nombre,1), nombre\_sede)),'@alumnoinacap.cl'),'A','A.') CORREO

FROM alumno.alumnos, inacap.sedes;

1. Utilizando columnas numéricas (que no sean claves primarias ni foráneas), realizar consultas con las funciones Round y Trunc, que también incluyan en su salida, el valor original.
   1. Mostrando dos decimales

SELECT superficie,

round(superficie/3,2) "un tercio de la superficie"

from gobierno.regiones;

* 1. Mostrando solo la parte entera

SELECT poblacion,

TRUNC(poblacion/2,0) "La mitad de la poblacion entero"

FROM gobierno.ciudades

* 1. Ajustado a mostrar valores en miles (centenas, decenas y unidades deben estar en cero)

1. Trabajo con funciones de fecha.

Usando la fecha de nacimiento de cada integrante del grupo, mostrar su nombre y apellido, la fecha de nacimiento (dia\_del\_mes/Mes\_en\_palabra/año\_en\_4\_digitos) la edad en años (entero) que tiene cada uno, la edad que tenía hace 7 meses, y la edad que tendrá en 10 meses más

* 1. BÁRBARA CARVAJAL

Select '13/agosto/1993' "Bárbara Carvajal",trunc((sysdate-to\_date('13/08/1993','dd/mm/yy'))/365) "EDAD",

trunc( (add\_months(sysdate,-7) - to\_date('13/08/1993','dd/mm/yy') )/365) "¿QUÉ EDAD TENÍA HACE 7 MESES?",

trunc( (add\_months(sysdate,10) - to\_date('13/08/1993','dd/mm/yy') )/365) "¿QUÉ EDAD TENDRÉ EN 7 MESES?"

from dual

* 1. Mi pololo

Select '02/enero/1993' "Camilo Monsalves",trunc((sysdate-to\_date('02/01/1995','dd/mm/yy'))/365) "EDAD",

trunc( (add\_months(sysdate,-7) - to\_date('02/01/1995','dd/mm/yy') )/365) "¿QUÉ EDAD TENÍA HACE 7 MESES?",

trunc( (add\_months(sysdate,10) - to\_date('02/01/1995','dd/mm/yy') )/365) "¿QUÉ EDAD TENDRÉ EN 7 MESES?"

from dual

* 1. Mi gatita

Select '13/febrero/2019' "Mokita",trunc((sysdate-to\_date('13/02/2019','dd/mm/yy'))/365) "EDAD",

trunc( (add\_months(sysdate,-7) - to\_date('13/02/2019','dd/mm/yy') )/365) "¿QUÉ EDAD TENÍA HACE 7 MESES?",

trunc( (add\_months(sysdate,10) - to\_date('13/02/2019','dd/mm/yy') )/365) "¿QUÉ EDAD TENDRÉ EN 7 MESES?"

from dual

1. Genere consultas de las funciones de fecha especificadas a continuación, usando columnas del tipo Date, de usuario alumno del modelo de datos de este trabajo, o del usuario HR del modelo ejemplo usado, o con las funciones sysdate o current\_timestamp
   1. Months\_between

SELECT trunc(months\_between(sysdate ,FECHA\_INICIO)) "meses de la carrera y fecha actual"

from alumno.historial\_carreras;

* 1. Add\_months

select fecha\_nace nacimiento,

add\_months(fecha\_nace, 10\*12) "cumplió 10 años el:"

from alumno.alumnos;

* 1. Next\_day

SELECT inicio\_estudio,

NEXT\_DAY(inicio\_estudio,'LUN') "PRÓXIMO LUNES"

FROM alumno.alumnos;

* 1. Last\_day

SELECT LAST\_DAY(FECHA\_FIN) "último día de la carrera"

FROM alumno.historial\_carreras;

1. Usando la función Trunc, obtener la fecha del primer día del mes en que nació uno de los integrantes del grupo. Mostrar el nombre completo, fecha de nacimiento, fecha del primer día del mes en que nació, y por último el día de la semana en palabra que corresponde a esa última fecha

select 'Bárbara Carvajal' "Nombre",

'13/agosto/1993' "Fecha de nacimiento",

trunc((sysdate-to\_date('13/08/1993', 'dd/mm/yy'))/365)"EDAD",

trunc(to\_date('13-AGOSTO-1993', 'DD-MON-YY'), 'MM') "PRIMER DÍA DEL MES",

TO\_CHAR(to\_date('13-AGOSTO-1993', 'DD-MON-YY'), 'DAY', 'NLS\_DATE\_LANGUAGE=SPANISH') "NACIÓ UN DÍA:"

from dual;

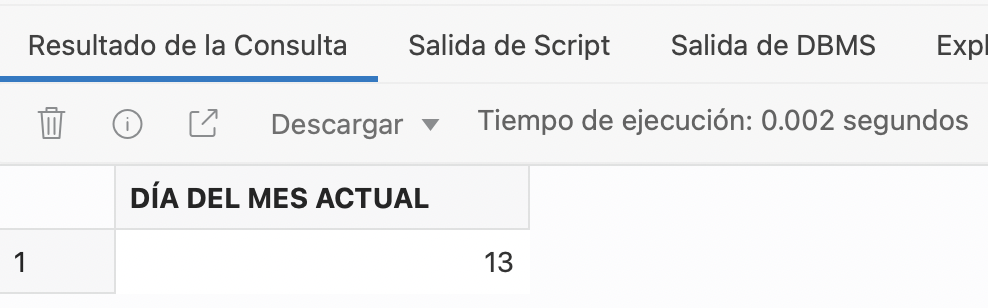
1. Usando la función Extract, mostrar en diferentes columnas los siguientes datos, según la fecha actual en el servidor de la base de datos: Día del mes, número del mes, año, hora, minuto, segundo (*Incluir en esta respuesta una imagen del resultado de la consulta en su máquina*)

-- día del mes

SELECT

EXTRACT(DAY from sysdate) "DÍA DEL MES ACTUAL"

from dual;

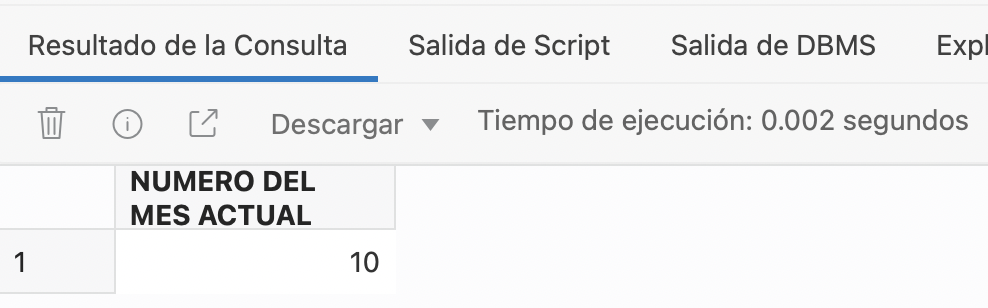


-- mes actual

SELECT

EXTRACT(MONTH from sysdate) "NUMERO DEL MES ACTUAL"

from dual;

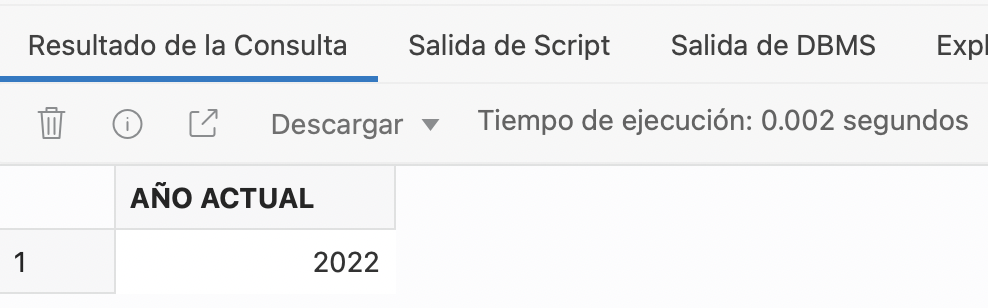


-- año actual

SELECT

EXTRACT(YEAR from sysdate) "AÑO ACTUAL"

from dual;

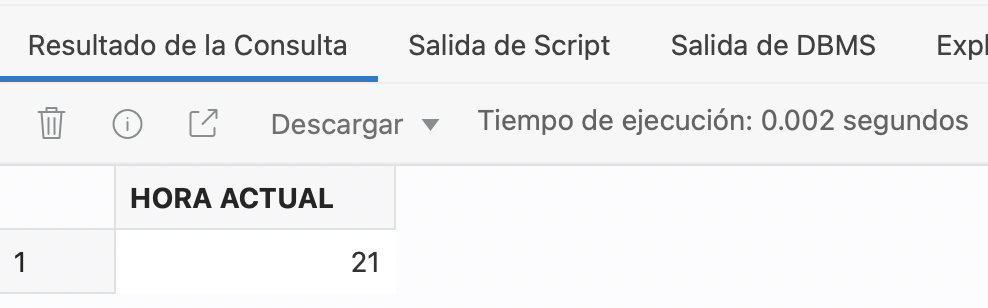


-- hora actual

SELECT

EXTRACT(Hour from systimestamp) "HORA ACTUAL"

from dual;

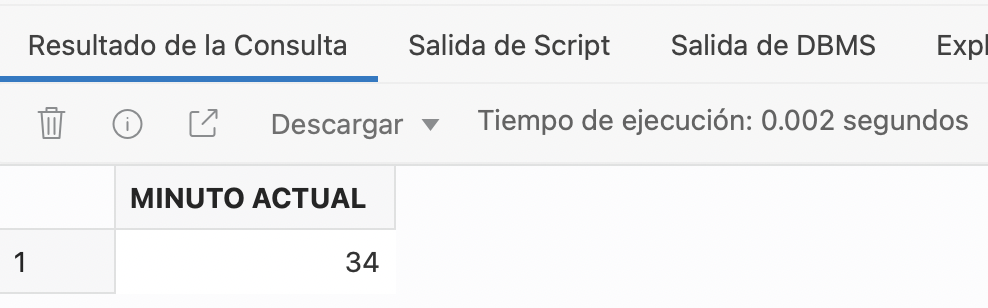


-- minuto actual

SELECT

EXTRACT(Minute from systimestamp) "MINUTO ACTUAL"

from dual;

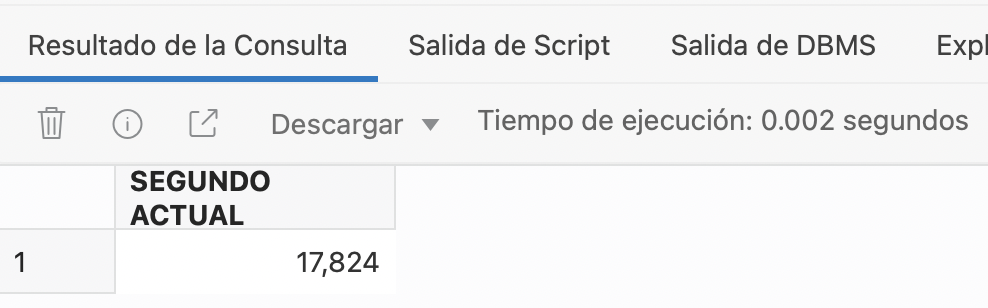


-- segundo actual

SELECT

EXTRACT(Second from systimestamp) "SEGUNDO ACTUAL"

from dual;



1. Para cada integrante del grupo, mostrar el texto como se muestra en el ejemplo (en la consulta solo puede ingresar la fecha de nacimiento, las otras fechas deben calcularse por medio de funciones y conversiones). (*Incluir en esta respuesta las imágenes de los resultados de la consulta en su máquina*)



Bárbara Carvajal:

select '13/agosto/1993' "Fecha nacimiento",

'Bárbara Carvajal' || ' ' || 'tendra un año mas el '||

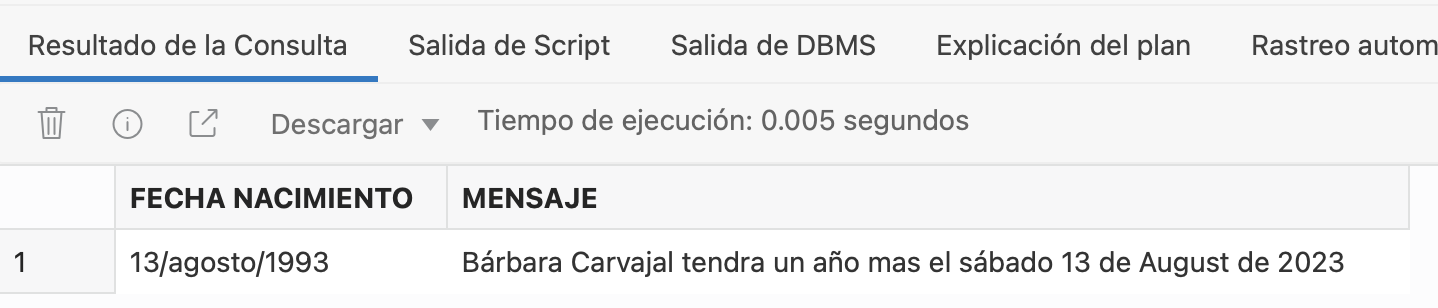
to\_char(to\_date('13/08/1993','dd/mm/yy')+365, 'day','NLS\_DATE\_LANGUAGE=SPANISH') ||

extract(DAY from to\_date('13/08/1993','dd/mm/yy')+365 )|| ' de ' ||

to\_char(to\_date('13/08/1993','dd/mm/yy'), 'Month') || ' de ' ||

EXTRACT(YEAR FROM sysdate+365) "Mensaje"

from dual;



Mi pololo:

select '02/enero/1995' "Fecha nacimiento",

'Camilo Monsalves' || ' ' || 'tendra un año mas el '||

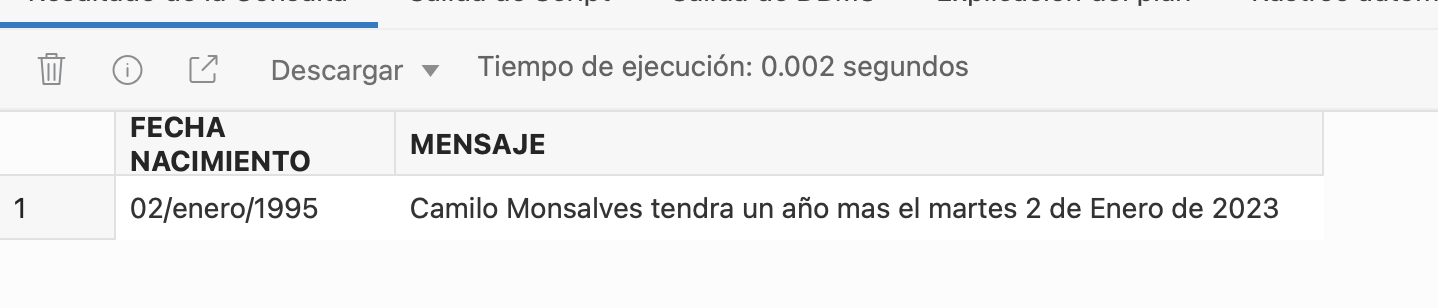
to\_char(to\_date('02/01/1995','dd/mm/yy')+365, 'day','NLS\_DATE\_LANGUAGE=SPANISH') ||

extract(DAY from to\_date('02/01/1995','dd/mm/yy')+365 )|| ' de ' ||

to\_char(to\_date('02/01/1995','dd/mm/yy'), 'Month') || ' de ' ||

EXTRACT(YEAR FROM sysdate+365) "Mensaje"

from dual;



Mi gatita:

select '13/febrero/2019' "Fecha nacimiento",

'Mokita' || ' ' || 'tendra un año mas el '||

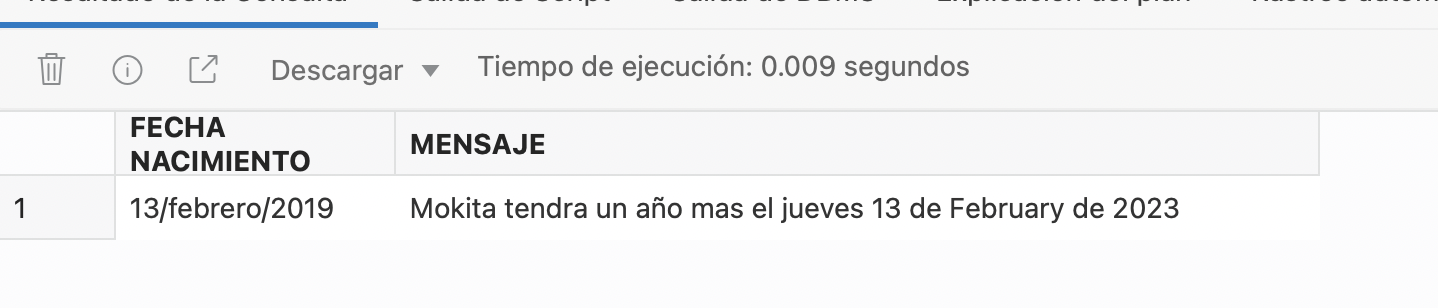
to\_char(to\_date('13/02/2019','dd/mm/yy')+365, 'day','NLS\_DATE\_LANGUAGE=SPANISH') ||

extract(DAY from to\_date('13/02/2019','dd/mm/yy')+365 )|| ' de ' ||

to\_char(to\_date('13/02/2019','dd/mm/yy'), 'Month') || ' de ' ||

EXTRACT(YEAR FROM sysdate+365) "Mensaje"

from dual;



1. Mostrar para 5 carreras del esquema Inacap del modelo, el nombre de la carrera, los semestres de la carrera, el valor del semestre, el valor mensual y el valor total de la carrera (un semestre tiene 5 meses). Todo campo con valores, debe tener un formato de moneda, colocando el símbolo de peses, separar los miles y establecer 1 decimal a mostrar.

SELECT NOMBRE\_CARRERA "CARRERA",

SEMESTRES "SEMESTRES ",

TO\_CHAR(VALOR\_SEMESTRE, '$9,999,999.9')"VALOR SEMESTRAL" ,

TO\_CHAR((VALOR\_SEMESTRE/5), '$9,999,999.9') "MENSUAL",

TO\_CHAR((VALOR\_SEMESTRE \* SEMESTRES), '$99,999,999.9' )

FROM inacap.CARRERAS

WHERE CARRERA\_ID IN('gasint', 'turism', 'tecenf', 'eleind', 'anapro' );

1. Uso de funciones generales para el tratamiento de valores nulos

Genere una consulta usando las tablas del modelo de datos, en las cuales se consideren el uso de las siguientes funciones:

* 1. Nvl()
  2. Nvl2()

--Compara si el alumno tiene tutor o no,

--la consulta devolver un si o un no dependiendo de si tienen nulos en el campo

select nombre, apellido,

NVL2(alumno\_tutor, 'SI', 'NO' ) CONDICIONAL

from alumno.alumnos

ORDER BY apellido;

* 1. Nullif()

--compara largo de caracteres de los nombres de alumno, si son

--iguales mostrara null,

--sino en el resultado mostrara la primera expresión

SELECT nombre,

LENGTH(nombre) "largo 1",

apellido,

LENGTH(apellido) "largo 2",

NULLIF(LENGTH(nombre),

LENGTH(apellido)) resultado

FROM alumno.alumnos;

* 1. Coalesce()

-- nos mostrará si un alumno tiene o no tutor

SELECT NOMBRE NOMBRE,

APELLIDO APELLIDO,

ALUMNO\_TUTOR TUTOR,

COALESCE(TO\_CHAR(ALUMNO\_TUTOR),

TO\_CHAR(NOMBRE)) "ALUMNOS SIN TUTOR"

FROM ALUMNOS;

1. Uso de expresiones condicionales
   1. ***Uso de Case***. Genere una clasificación del tipo de sede, según la cantidad de alumno que tiene (por ejemplo, si tiene entre 0 y 10, podría decir que es una sede *Crítica*), usando la expresión condicional CASE. Mostrar “todas las sedes”, la cantidad de alumnos que tiene, y su clasificación. (considere al mínimo 5 tipos de clasificaciones distintas)
   2. ***Uso de Decode***. Genere una consulta, distinta a la anterior, que utilice DECODE en una consulta que al menos considere dos tablas del modelo. Debe tener al menos 5 opciones distintas a considerar en la condición en Decode.

1. Concat(), Substr(), Length(), Instr(), Lpad(), Rpad(), Replace() [↑](#footnote-ref-1)