

---

## 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

## A. Introducción

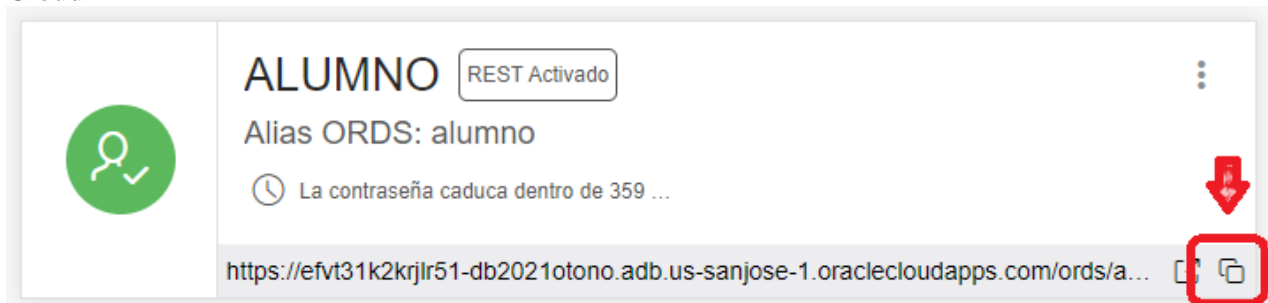
Para el siguiente trabajo en grupo, ustedes deberán tener creada una cuenta en Oracle Cloud (*Referencias de Creación en el Ambiente de Aprendizaje Inacap*), usando las estructuras desarrolladas en la primera actividad asincrónica de la asignatura

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)

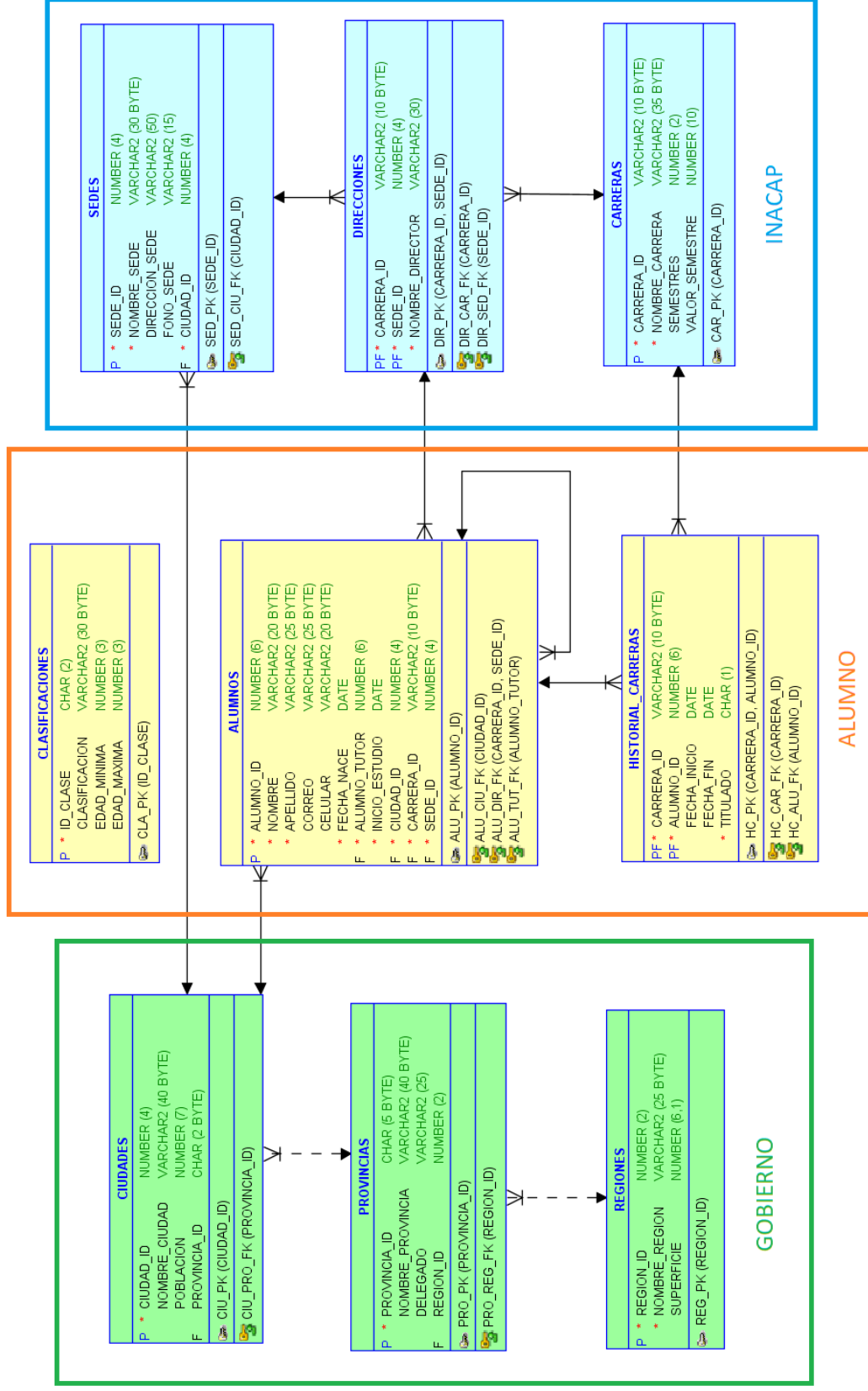
(\*) **IMPORTANTE:** para cada respuesta dada, deben indicar quien es el integrante del grupo que generó la respuesta, en caso de requerir consulta por parte del docente (debe estar equilibrado entre los integrantes)

## B. 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



## C. 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.

Uniéndolo 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:

#### 1.1. 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;
```

#### 1.2. 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";
```

#### 1.3. 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";
```

#### 1.4. 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;
```

#### 1.5. 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;
```

## 2. Uso de funciones de Carácter

2.1. 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_CIUADAD = 'Coquimbo';
```

2.2. Usando las tablas en el esquema Gobierno, realice 3 consultas usando funciones de manipulación<sup>1</sup> de caracteres (cada consulta utilizando dos funciones distintas)

2.2.1.

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

```
SELECT nombre_region ,
lower(substr (nombre_region,-1))
from gobierno.regiones;
```

2.2.2.

-- 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;
```

2.2.3.

--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;
```

2.3. Usando una de las tablas en el esquema Alumno, realizar una consulta que anide tres funciones de manipulación<sup>1</sup> de caracteres

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

---

<sup>1</sup> Concat(), Substr(), Length(), Instr(), Lpad(), Rpad(), Replace()

-- 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, incap.sedes;
```

3. 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.

#### 3.1. Mostrando dos decimales

```
SELECT superficie,  
round(superficie/3,2) "un tercio de la superficie"  
from gobierno.regiones;
```

#### 3.2. Mostrando solo la parte entera

```
SELECT poblacion,  
TRUNC(poblacion/2,0) "La mitad de la poblacion entero"  
FROM gobierno.ciudades
```

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

4. 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

#### 4.1. 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
```

#### 4.2. 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
```

#### 4.3. 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
```

5. 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

##### 5.1. Months\_between

```
SELECT trunc(months_between(sysdate ,FECHA_INICIO)) "meses de la carrera y fecha actual"  
from alumno.historial_carreras;
```

##### 5.2. Add\_months

```
select fecha_nace nacimiento,  
add_months(fecha_nace, 10*12) "cumplió 10 años el:"  
from alumno.alumnos;
```

##### 5.3. Next\_day

```
SELECT inicio_estudio,
```



```
NEXT_DAY(inicio_estudio,'LUN') "PRÓXIMO LUNES"
FROM alumno.alumnos;
```

#### 5.4. Last\_day




```
SELECT LAST_DAY(FECHA_FIN) "último día de la carrera"
FROM alumno.historial_carreras;
```

6. 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;
```

7. 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;
```

Resultado de la Consulta	Salida de Script	Salida de DBMS	Expl
   Descargar ▼ Tiempo de ejecución: 0.002 segundos			
DÍA DEL MES ACTUAL			
1		13	

```
-- mes actual
```

SELECT




```
EXTRACT(MONTH from sysdate) "NUMERO DEL MES ACTUAL"  
from dual;
```

Resultado de la Consulta		Salida de Script	Salida de DBMS	Expl
  		Descargar ▼	Tiempo de ejecución: 0.002 segundos	
NUMERO DEL MES ACTUAL				
1	10			

-- año actual

SELECT




```
EXTRACT(YEAR from sysdate) "AÑO ACTUAL"  
from dual;
```

Resultado de la Consulta		Salida de Script	Salida de DBMS	Expl
<div><div></div><div>Descargar ▼</div></div>		Tiempo de ejecución: 0.002 segundos		
	AÑO ACTUAL			
1	2022			

-- hora actual

SELECT

```
EXTRACT(Hour from systimestamp) "HORA ACTUAL"  
from dual;
```




Resultado de la Consulta		Salida de Script	Salida de DBMS	Expl
  		Descargar ▼	Tiempo de ejecución: 0.002 segundos	
	HORA ACTUAL			
1	21			

-- minuto actual

SELECT

EXTRACT(Minute from systimestamp) "MINUTO ACTUAL"

from dual;




Resultado de la Consulta		Salida de Script	Salida de DBMS	Expl
  		Descargar ▼	Tiempo de ejecución: 0.002 segundos	
	MINUTO ACTUAL			
1	34			

-- segundo actual

SELECT

EXTRACT(Second from systimestamp) "SEGUNDO ACTUAL"

from dual;

Resultado de la Consulta		Salida de Script	Salida de DBMS	Expl
  		Descargar ▼	Tiempo de ejecución: 0.002 segundos	
	SEGUNDO ACTUAL			
1	17,824			

8. 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)

Fecha Nace	MENSAJE
13/01/95	Juan Perez tendrá un año más el <b>Jueves 13 de Enero de 2022</b>



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;
```

	FECHA NACIMIENTO	MENSAJE
1	13/agosto/1993	Bárbara Carvajal tendra un año mas el sábado 13 de August de 2023

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;
```

	FECHA NACIMIENTO	MENSAJE
1	02/enero/1995	Camilo Monsalves tendra un año mas el martes 2 de Enero de 2023

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;
```

	FECHA NACIMIENTO	MENSAJE
1	13/febrero/2019	Mokita tendra un año mas el jueves 13 de February de 2023

9. 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 pesos, separar los miles y establecer 1 decimal a mostrar.

```
SELECT NOMBRE_CARRERA "CARRERA",
       SEMESTRES "SEMESTRES ",
       TO_CHAR(VAOR_SEMESTRE, '$9,999,999.9') "VALOR SEMESTRAL" ,
       TO_CHAR((VAOR_SEMESTRE/5), '$9,999,999.9') "MENSUAL",
       TO_CHAR((VAOR_SEMESTRE * SEMESTRES), '$99,999,999.9' )
FROM inacap.CARRERAS
WHERE CARRERA_ID IN('gasint', 'turism', 'tecenf', 'eleind', 'anapro');
```

10. 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:

10.1. NVL()

10.2. 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;
```

10.3. 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;
```

#### 10.4. 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;
```

### 11. Uso de expresiones condicionales

- 11.1. **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)
- 11.2. **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.