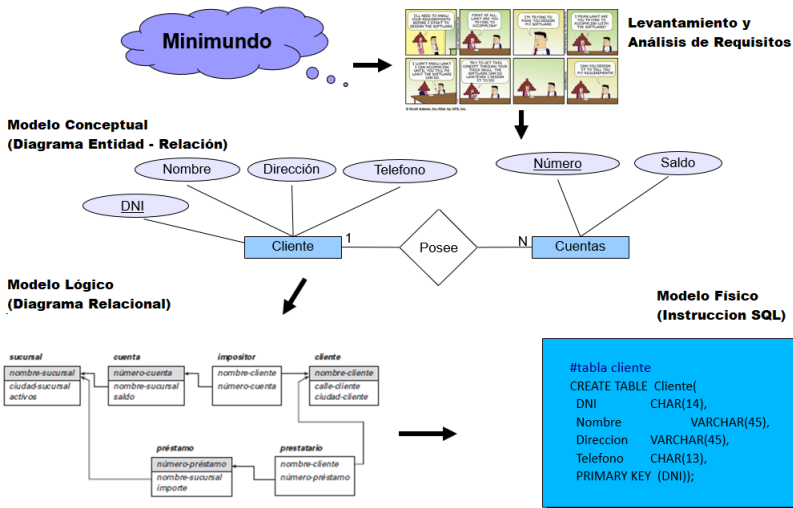


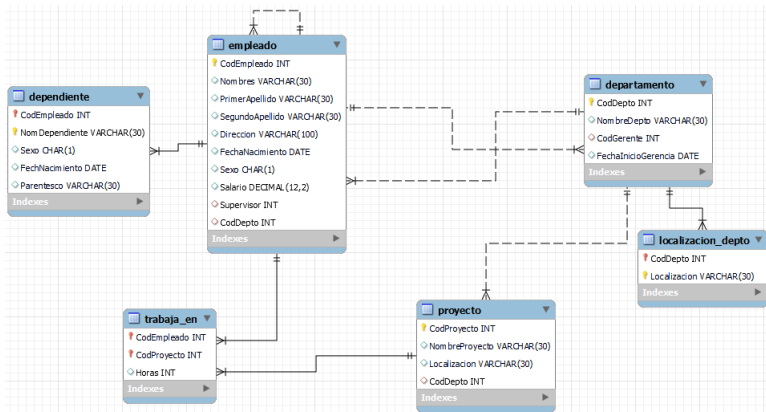
Índice

- 1 Fases del Proyecto de BD
- 2 Lenguaje de Manipulación de Datos en BDR
- 3 Funciones MYSQL - Agregación
- 4 Funciones MYSQL - Cadenas de Caracteres
- 5 Funciones MYSQL - Numéricas
- 6 Funciones MYSQL - Fechas
- 7 MYSQL - Otras Funciones o Cláusulas

Fases del Proyecto de BD



Base de Datos Empresa



Base de Datos Empresa

CodEmpleado	Nombres	PrimerApellido	SegundoApellido	Direccion	FechaNacimiento	Sexo	Salario	Supervisor	CodDepto
123456789	Juan	Perez	Rodriguez	Calle Numero A 1	1965-01-09	M	300.00	333445555	5
333445555	Frank	Velazquez	Flores	Calle Numero B 2	1955-12-08	M	4000.00	888665555	5
453453453	Daniela	Acco	Olvarez	Calle Numero F 6	1962-07-31	F	2500.00	333445555	5
666884444	Pedro	Lima	Maldonado	Calle Numero E 5	1952-09-15	M	1200.00	333445555	5
888665555	Francisco	Linares	Gomez	Calle Numero H 8	1957-11-10	M	5500.00	NULL	1
987654321	Luisa	Santos	Ferrel	Calle Numero D 4	1951-06-20	F	430.00	888665555	4
987987987	Mateo	Vela	Marruecos	Calle Numero G 7	1979-03-29	M	2500.00	987654321	4
999887777	Alice	Jimenez	Portugal	Calle Numero C 3	1968-07-19	F	2500.00	987654321	4

Base de Datos Empresa

CodDepto	NombreDepto	CodGerente	FechaInicioGerencia
1	Direccion	888665555	2001-06-19
4	Administracion	987654321	1995-01-01
5	Investigacion	333445555	1998-05-22

Base de Datos Empresa

CodEmpleado	NomDependiente	Sexo	FechNacimiento	Parentesco
123456789	Anabel	F	1998-12-31	Hija
123456789	Elizabeth	F	1957-05-05	Conyugue
123456789	Michael	M	1988-01-21	Hijo
333445555	Ana	F	1967-04-03	Hija
333445555	Juana	F	1958-05-05	Conyugue
333445555	Victor	M	1973-10-25	Hijo
987654321	Igor	M	1952-02-29	Conyugue

Base de Datos Empresa

CodDepto	Localizacion
1	Lima
4	Arequipa
5	Cusco
5	Puno
5	Trujillo

Base de Datos Empresa

CodProyecto	NombreProyecto	Localizacion	CodDepto
1	Proyecto X	Oficina 317	5
2	Proyecto Y	Oficina 22	5
3	Proyecto Z	Oficina 544	5
10	Proyecto A	Oficina 51	4
20	Proyecto B	Oficina 67	1
30	Proyecto C	Oficina 385	4

Base de Datos Empresa

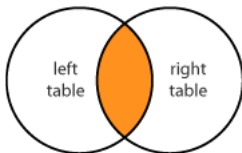
CodEmpleado	CodProyecto	Horas
123456789	1	32
123456789	2	7
333445555	2	10
333445555	3	10
333445555	10	10
333445555	20	10
453453453	1	20
453453453	2	20
666884444	3	40
888665555	20	7
987654321	20	15
987654321	30	20
987987987	10	35
987987987	30	5
999887777	10	10
999887777	30	30

Lenguajes de consulta en BDR

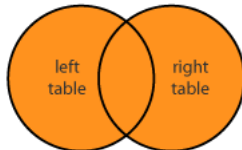
- Algebra Relacional.
- Cálculo Relacional.
- SQL (Structured Query Language).

SQL - INNER, FULL, LEFT, RIGHT JOIN

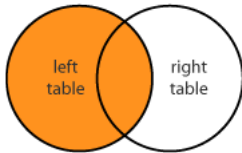
INNER JOIN



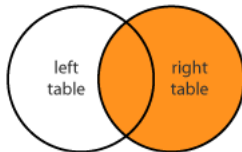
FULL JOIN



LEFT JOIN



RIGHT JOIN

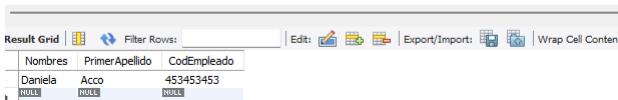


SQL - LIMIT

- La cláusula LIMIT se utiliza para especificar el número de registros a devolver.
- La cláusula LIMIT es útil en resultados con una gran cantidad de registros. Devolver una gran cantidad de registros puede afectar el rendimiento.
- Ejemplo: Muestre los nombres, primer apellido y código de la primer empleada mujer según el ordenamiento del primer apellido.

SQL - LIMIT

```
1 • USE empresa;  
2 • SELECT  
3     Nombres, PrimerApellido, CodEmpleado FROM empleado  
4     WHERE Sexo = 'F'  
5     ORDER BY PrimerApellido  
6     LIMIT 1
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content

	Nombres	PrimerApellido	CodEmpleado
1	Daniela	Acco	453453453
2	NULL	NULL	NULL

SQL - MIN

- La función MIN() retorna el menor valor de una columna.
- Es una función de agregación.
- Ejemplo: Muestre el menor salario de todos los empleados.

SQL - MIN

```
1 • USE empresa;  
2 • SELECT MIN(Salario) AS 'Menor Salario'  
3   FROM empleado;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Menor Salario
	300.00

SQL - MAX

- La función MAX() retorna el mayor valor de una columna.
- Es una función de agregación.
- Ejemplo: Muestre el mayor salario de todos los empleados.

SQL - MAX

- 1 • `USE empresa;`
- 2 • `SELECT MAX(Salario) AS 'Mayor Salario'`
- 3 • `FROM empleado;`





Result Grid				Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Mayor Salario					
▶	5500.00					

SQL - COUNT

- La función COUNT() retorna el número de filas que coinciden con una condición.
- Es una función de agregación.
- Ejemplo: Muestre la cantidad de departamentos que existen.

SQL - COUNT


- 1 • `USE empresa;`
- 2 • `SELECT COUNT(*) AS 'Cantidad Departamentos'`
- 3 • `FROM departamento;`

Result Grid				Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Cantidad Departamentos					
▶	3					

SQL - COUNT

- Ejemplo: Muestre la cantidad de empleadas mujeres en la empresa

```
1 • USE empresa;  
2 • SELECT COUNT(CodEmpleado) AS 'Cantidad Empleadas Mujeres'  
3   FROM empleado  
4   WHERE Sexo = 'F';
```



Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Cantidad Empleadas Mujeres				
3				

SQL - AVG

- La función `AVG()` retorna el promedio de una columna numérica.
- Es una función de agregación.
- Ejemplo: Muestre el promedio de sueldo de los empleados hombres.

SQL - AVG

- 1 • `USE empresa;`
- 2 • `SELECT AVG(Salario) AS 'Promedio sueldo hombres'`
- 3 `FROM empleado`
- 4 `WHERE Sexo = 'M';`

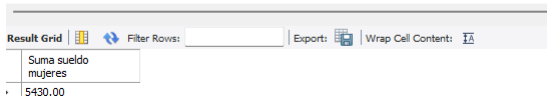
Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Promedio sueldo hombres				
2700.000000				

SQL - SUM

- La función SUM() retorna la suma de una columna numérica.
- Es una función de agregación.
- Ejemplo: Muestre la suma de sueldo de las empleadas mujeres.

SQL - SUM

```
1 • USE empresa;  
2 • SELECT SUM(Salario) AS 'Suma sueldo mujeres'  
3   FROM empleado  
4  WHERE Sexo = 'F';
```








Suma sueldo mujeres
5430.00

SQL - GROUP BY

- La clausula GROUP BY agrupa filas que tienen el mismo valor en una sola línea.
- Se utiliza frecuentemente con las funciones de agregación.
- Ejemplo: Muestre la cantidad de empleados, la suma de salarios y promedio de salarios de empleados hombres y mujeres.

SQL - GROUP BY

```
1 • USE empresa;  
2 • SELECT sexo,  
3     COUNT(CodEmpleado) 'Cantidad de empleados',  
4     SUM(Salario) 'Suma salarios',  
5     AVG(Salario) 'Promedio salarios'  
6 FROM empleado  
7 GROUP BY(Sexo);
```

Result Grid				
			 Filter Rows: <input type="text"/>	Export: 
		Wrap Cell Content:  		
sexo	Cantidad de empleados	Suma salarios	Promedio salarios	
M	5	13500.00	2700.000000	
F	3	5430.00	1810.000000	

SQL - GROUP BY

- Ejemplo: Muestre el código, nombre, primer apellido y cantidad de dependientes por cada empleado.

```
1 • USE empresa;  
2 SELECT e.CodEmpleado, e.Nombres, e.PrimerApellido, COUNT(de.CodEmpleado)  
3 FROM empleado e  
4 INNER JOIN dependiente de  
5 ON e.CodEmpleado = de.CodEmpleado  
6 GROUP BY e.CodEmpleado, e.Nombres, e.PrimerApellido  
7
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content:

CodEmpleado	Nombres	PrimerApellido	COUNT(de.CodEmpleado)
123456789	Juan	Perez	3
333445555	Frank	Velazquez	3
987654321	Luisa	Santos	1

SQL - HAVING

- La clausula HAVING es similar a la cláusula WHERE pero para funciones de agregación.
- Ejemplo: Muestre la cantidad de empleados, la suma de salarios y promedio de salarios de empleados hombres y mujeres donde la cantidad de empleados sea menor que 4.

SQL - HAVING

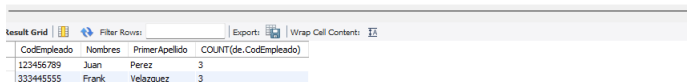
```
1 • USE empresa;  
2 SELECT sexo,  
3 COUNT(CodEmpleado) 'Cantidad de empleados',  
4 SUM(Salario) 'Suma salarios',  
5 AVG(Salario) 'Promedio salarios'  
6 FROM empleado  
7 GROUP BY(Sexo)  
8 HAVING COUNT(CodEmpleado) < 4;  
9
```

Result Grid				
		Filter Rows:	Export:	Wrap Cell Content:
sexo	Cantidad de empleados	Suma salarios	Promedio salarios	
F	3	5430.00	1810.000000	

SQL - HAVING

- Ejemplo: Muestre el código, nombre, primer apellido y cantidad de dependientes por cada empleado donde la cantidad de dependientes sea mayor o igual a 3.

```
1 • USE empresa;  
2 SELECT e.CodEmpleado, e.Nombres, e.PrimerApellido, COUNT(de.CodEmpleado)  
3 FROM empleado e  
4 INNER JOIN dependiente de  
5 ON e.CodEmpleado = de.CodEmpleado  
6 GROUP BY e.CodEmpleado, e.Nombres, e.PrimerApellido  
7 HAVING COUNT(de.CodEmpleado) >= 3;  
8  
9
```



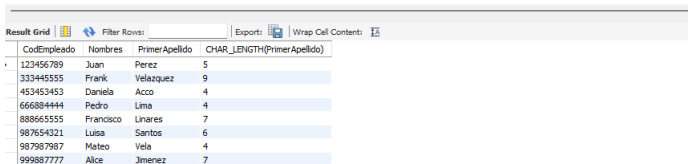
CodEmpleado	Nombres	PrimerApellido	COUNT(de.CodEmpleado)
123456789	Juan	Perez	3
333445555	Frank	Velazquez	3

SQL - CHAR_LENGTH

- La función CHAR_LENGTH() retorna el tamaño de una cadena de texto.
- Es igual a la función CHARACTER_LENGTH()
- Ejemplo: Muestre el código, nombre, primer apellido y la cantidad de caracteres del primer apellido para cada empleado.

SQL - CHAR_LENGTH

- 1 • **USE** empresa;
- 2 **SELECT** CodEmpleado, Nombres, PrimerApellido, CHAR_LENGTH(PrimerApellido)
- 3 **FROM** empleado



CodEmpleado	Nombres	PrimerApellido	CHAR_LENGTH(PrimerApellido)
123456789	Juan	Perez	5
333445555	Frank	Velazquez	9
453453453	Daniela	Acco	4
666884444	Pedro	Lima	4
888665555	Francisco	Linares	7
987654321	Luisa	Santos	6
987987987	Mateo	Vela	4
999887777	Alice	Jimenez	7

SQL - CONCAT

- La función `CONCAT()` concatena dos o más cadenas de texto.
- Ejemplo: Muestre el código, primer apellido, segundo apellido y nombre (los tres últimos campos en un solo campo); y el sueldo de todas las empleadas mujeres.

SQL - CONCAT

```
1 • USE empresa;  
2 SELECT CodEmpleado,  
3 CONCAT(PrimerApellido, ' ', SegundoApellido, ' ', Nombres) AS 'Apellidos y Nombres',  
4 Salario  
5 FROM empleado  
6 WHERE Sexo = 'F'
```

Result Grid			
Filter Rows:		Exports:	Wrap Cell Content: I1
CodEmpleado	Apellidos y Nombres	Salario	
453453453	Acco Olvarez, Daniela	2500.00	
987654321	Santos Ferrel, Luisa	430.00	
999887777	Jimenez Portugal, Alice	2500.00	

SQL - CONCAT_WS

- La función `CONCAT_WS()` concatena dos o más cadenas de texto con un separador.
- Ejemplo: Muestre el código, primer apellido, segundo apellido y nombre (los tres últimos campos en un solo campo); y el sueldo de todos los empleados hombres.

SQL - CONCAT_WS

```
1 • USE empresa;
2   SELECT CodEmpleado,
3   CONCAT_WS("/", PrimerApellido, SegundoApellido, Nombres) AS 'Apellidos y Nombres',
4   Salario
5   FROM empleado
6   WHERE Sexo = 'M'
```

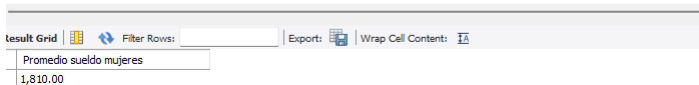
Result Grid			Filter Rows:	Exports:	Wrap Cell Content:
CodEmpleado	Apellidos y Nombres	Salario			
123456789	Perez/Rodriguez/Juan	300.00			
333445555	Velazquez/Flores/Frank	4000.00			
666884444	Lima/Maldonado/Pedro	1200.00			
888665555	Linares/Gomez/Francisco	5500.00			
987987987	Vela/Marruecos/Mateo	2500.00			

SQL - FORMAT

- La función `FORMAT()` asigna el formato "`#,###,###.##`" a un número redondeado a una cantidad de decimales.
- Ejemplo: Muestre el salario promedio de las empleadas mujeres en el formato "`#,###,###.##`" con 2 decimales.

SQL - FORMAT

```
1 • USE empresa;  
2 SELECT FORMAT(AVG(Salario),2) AS 'Promedio sueldo mujeres'  
3 FROM empleado  
4 WHERE Sexo = 'F';
```





Promedio sueldo mujeres
1,810.00

SQL - INSERT

- La función `INSERT()` inserta una cadenas de caracteres dentro de otra cadenas de caracteres en una determinada posición y en un cierto número de caracteres.
- `INSERT(string, position, number, string2)`
- Ejemplo: Muestre el código, nombre y localización de cada proyecto, pero en vez de "Oficina" mostrar "Of.".

SQL - INSERT

- 1 • **USE** empresa;
- 2 • **SELECT** CodProyecto, NombreProyecto,
- 3 **INSERT**(Localizacion, 3, 5, ".") Oficina
- 4 **FROM** proyecto

Result Grid			
		Filter Rows:	Export:  Wrap Cell Content: 
	CodProyecto	NombreProyecto	Oficina
▶	1	Proyecto X	Of. 317
	2	Proyecto Y	Of. 22
	3	Proyecto Z	Of. 544
	10	Proyecto A	Of. 51
	20	Proyecto B	Of. 67
	30	Proyecto C	Of. 385

SQL - IN

- La función IN() permite especificar varios valores para una cláusula WHERE.
- La función IN() es un forma reducida de realizar multiples condiciones OR.
- Ejemplo: Muestre el código, nombre y localización de cada proyecto con nombre "Proyecto X" o "Proyecto Y" o "Proyecto Z".

SQL - IN

- 1 • **USE** empresa;
- 2 • **SELECT** CodProyecto, NombreProyecto, Localizacion
- 3 **FROM** proyecto
- 4 **WHERE** NombreProyecto **IN** ('Proyecto X', 'Proyecto Y', 'Proyecto Z')

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
CodProyecto	NombreProyecto	Localizacion				
1	Proyecto X	Oficina 317				
2	Proyecto Y	Oficina 22				
3	Proyecto Z	Oficina 544				
NULL	NULL	NULL				

SQL - LCASE

- La función LCASE() convierte una cadenas de texto a minúsculas.
- Ejemplo: Muestre el código, nombre en minúsculas y localización en minúsculas de cada proyecto.

SQL - LCASE

```
1 • USE empresa;  
2 • SELECT CodProyecto,  
3     LCASE(NombreProyecto) 'Proyecto',  
4     LCASE(Localizacion) 'Localización'  
5 • FROM proyecto
```

Result Grid				Filter Rows:		Export:	Wrap Cell Content
	CodProyecto	Proyecto	Localización				
1		proyecto x	oficina 317				
2		proyecto y	oficina 22				
3		proyecto z	oficina 544				
10		proyecto a	oficina 51				
20		proyecto b	oficina 67				
30		proyecto c	oficina 385				

SQL - UCASE

- La función UCASE() convierte una cadenas de texto a mayúsculas.
- Ejemplo: Muestre el código, nombre en mayúsculas y localización en mayúsculas de cada proyecto.

SQL - UCASE

```
1 • USE empresa;  
2 • SELECT CodProyecto,  
3     UCASE(NombreProyecto) 'Proyecto',  
4     UCASE(Localizacion) 'Localización'  
5 • FROM proyecto
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content
CodProyecto	Proyecto	Localización			
1	PROYECTO X	OFICINA 317			
2	PROYECTO Y	OFICINA 22			
3	PROYECTO Z	OFICINA 544			
10	PROYECTO A	OFICINA 51			
20	PROYECTO B	OFICINA 67			
30	PROYECTO C	OFICINA 385			

SQL - LEFT

- La función LEFT() extrae un número de caracteres de una cadena de texto, empieza desde el primer caracter más a la izquierda.
- LEFT(string, number of chars)
- Ejemplo: Muestre el código, nombre, primer apellido y segundo apellido de todos los empleados cuyo primer apellido comienza con un vocal.

SQL - LEFT

```
1 • USE empresa;  
2 SELECT  
3 CodEmpleado, Nombres, PrimerApellido FROM empleado  
4 WHERE LEFT(UCASE(PrimerApellido), 1) IN ('A','E','I','O','U')
```

Result Grid				Filter Rows:				Edit:				Export/Import:				Wrap Cell Content:			
	CodEmpleado	Nombres	PrimerApellido																
	453453453	Daniela	Acco																

SQL - RIGHT

- La función RIGHT() extrae un número de caracteres de una cadena de texto, empieza desde el primer caracter más a la derecha.
- RIGHT(string, number of chars)
- Ejemplo: Muestre el código, nombre, primer apellido y segundo apellido de todos los empleados cuyo primer apellido termine en una consonante.

SQL - RIGHT

```
1 • USE empresa;  
2 SELECT  
3 CodEmpleado, Nombres, PrimerApellido FROM empleado  
4 WHERE RIGHT(LCASE(PrimerApellido), 1) NOT IN ('a','e','i','o','u')
```

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
CodEmpleado	Nombres	PrimerApellido				
123456789	Juan	Perez				
333445555	Frank	Velazquez				
888665555	Francisco	Linares				
987654321	Luisa	Santos				
999887777	Alice	Jimenez				

SQL - TRIM

- La función TRIM() elimina los espacios en blanco al comienzo y al final de una cadena de texto.
- Ejemplo: Muestre el código, primer apellido, segundo apellido y nombre (los tres últimos campos en un solo campo); y el sueldo de todas las empleadas mujeres.

SQL - TRIM

```
1 • USE empresa;  
2 SELECT CodEmpleado,  
3 TRIM(CONCAT(' ', PrimerApellido, ' ', SegundoApellido, ' ', Nombres, ' ')) AS 'Apellidos y Nombres',  
4 Salario  
5 FROM empleado  
6 WHERE Sexo = 'F'
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
CodEmpleado	Apellidos y Nombres	Salario			
453453453	Acco Olvarez, Daniele	2500.00			
987654321	Santos Pierrel, Luisa	430.00			
999887777	Jimenez Portugal, Alice	2500.00			

SQL - LTRIM

- La función LTRIM() elimina los espacios en blanco al comienzo o izquierda de una cadena de texto.
- Ejemplo: Muestre el código, primer apellido, segundo apellido y nombre (los tres últimos campos en un solo campo); y el sueldo de todas las empleadas mujeres.

SQL - LTRIM

```
1 • USE empresa;  
2 SELECT CodEmpleado,  
3 LTRIM(CONCAT(' ', PrimerApellido, ' ', SegundoApellido, ' ', Nombres, ' ')) AS 'Apellidos y Nombres',  
4 Salario  
5 FROM empleado  
6 WHERE Sexo = 'F'
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content: 12
CodEmpleado	Apellidos y Nombres	Salario			
453453453	Acos Olvarez, Daniela	2500.00			
987654321	Santos Ferrel, Luisa	430.00			
999887777	Jimenez Portugal, Alice	2500.00			

SQL - RTRIM

- La función LTRIM() elimina los espacios en blanco al final o derecha de una cadena de texto.
- Ejemplo: Muestre el código, primer apellido, segundo apellido y nombre (los tres últimos campos en un solo campo); y el sueldo de todas las empleadas mujeres.

SQL - RTRIM

```
1 • USE empresa;  
2 SELECT CodEmpleado,  
3 RTRIM(CONCAT(' ', PrimerApellido, ' ', SegundoApellido, ' ', Nombres, ' ')) AS 'Apellidos y Nombres',  
4 Salario  
5 FROM empleado  
6 WHERE Sexo = 'F'
```

Result Grid			Filter Rows:	Export:	Wrap Cell Contents:
CodEmpleado	Apellidos y Nombres	Salario			
453453453	Acce Olvarez, Daniela	2500.00			
987654321	Santos Ferrel, Luisa	430.00			
999887777	Jimenez Portugal, Alice	2500.00			

SQL - REPLACE

- La función REPLACE() reemplaza todas las ocurrencias de una sub cadena de texto en una cadena de texto con una nueva cadena de texto. REPLACE(string, substring, new_string)
- Ejemplo: Muestre el código, nombre y localización de cada proyecto, pero en vez de "Oficina" mostrar "Of.".

SQL - REPLACE

- 1 • **USE** empresa;
- 2 **SELECT** CodProyecto, NombreProyecto,
- 3 **REPLACE**(Localizacion, "Oficina", "Of.") Oficina
- 4 **FROM** proyecto

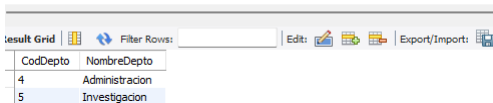
result Grid			Filter Rows:	Export:	Wrap Cell Content:
CodProyecto	NombreProyecto	Oficina			
1	Proyecto X	Of. 317			
2	Proyecto Y	Of. 22			
3	Proyecto Z	Of. 544			
10	Proyecto A	Of. 51			
20	Proyecto B	Of. 67			
30	Proyecto C	Of. 385			

SQL - POSITION

- La función POSITION() retorna la posición de la primera ocurrencia de una sub cadena de texto en una cadena de texto. Sino se encuentra el sub cadena de texto resultado es 0.
- Ejemplo: Muestre el código y nombre de cada departamento que en su nombre contenta la cadena de texto "acion".

SQL - POSITION

```
1 • USE empresa;  
2 SELECT CodDepto, NombreDepto  
3 FROM departamento  
4 WHERE POSITION("acion" IN NombreDepto) != 0
```



The screenshot shows a database interface with a 'result Grid' tab. Above the table, there are controls for 'Filter Rows', 'Edit', and 'Export/Import'. The table has two columns: 'CodDepto' and 'NombreDepto'. It contains two rows of data. The first row has '4' in the 'CodDepto' column and 'Administracion' in the 'NombreDepto' column. The second row has '5' in the 'CodDepto' column and 'Investigacion' in the 'NombreDepto' column. The second row is highlighted with a blue background.

CodDepto	NombreDepto
4	Administracion
5	Investigacion

SQL - STRCMP

- La función STRCMP() compara dos cadenas. El valor 0 indica que ambas cadenas son iguales, -1 o 1 si son diferentes.
- Ejemplo: Muestre el código, nombre, primer apellido y segundo apellido de cada empleado que tengan como nombre Juan o Pedro.

SQL - STRCMP

```
1 • USE empresa;  
2 SELECT CodEmpleado, Nombres, PrimerApellido, SegundoApellido  
3 FROM empleado  
4 WHERE STRCMP(Nombres,"Juan") = 0 OR STRCMP(Nombres,"Pedro") = 0
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:




	CodEmpleado	Nombres	PrimerApellido	SegundoApellido
	123456789	Juan	Perez	Rodriguez
	666884444	Pedro	Lima	Maldonado

SQL - SUBSTRING

- La función `SUBSTRING()` extrae una sub cadena de texto de una cadena de texto desde una posición y con una cantidad de caracteres.
- Las funciones `SUBSTR()` y `MID()` son iguales a la función `SUBSTRING()`.
- `SUBSTRING(string, start, length)`
- Ejemplo: Muestre el código, nombre y los dos primeros caracteres del primer apellido de cada empleado.

SQL - SUBSTRING

```
1 • USE empresa;  
2 SELECT CodEmpleado, Nombres, PrimerApellido,  
3 SUBSTRING(PrimerApellido, 1, 2) AS 'Dos primeros caracteres'  
4 FROM empleado  
5
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 			
CodEmpleado	Nombres	PrimerApellido	Dos primeros caracteres
123456789	Juan	Perez	Pe
333445555	Frank	Velazquez	Ve
453453453	Daniela	Acco	Ac
666884444	Pedro	Lima	Li
888665555	Francisco	Linares	Li
987654321	Luisa	Santos	Sa
987987987	Mateo	Vela	Ve
999887777	Alice	Jimenez	Ji

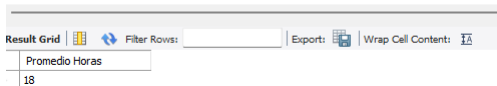
Funciones MYSQL - Numéricas

- Existen un variedad de funciones numéricas (Tarea), entre las principales:
 - ABS: Valor absoluto de un número.
 - CEILING, CEIL: Entero más pequeño no inferior a un número.
 - DIV: Resultado de la división entera de dos números.
 - FLOOR: Entero más grande no superior a un número.
 - MOD, %: Resto de la división entera de dos números.
 - RAND: Número aleatorio superior o igual a 0 y estrictamente inferior a 1.
 - ROUND: Número redondeado con la precisión solicitada.

Funciones MYSQL - Numéricas

- Ejemplo: Muestre el promedio sin decimales de la cantidad de horas que trabaja cada empleado en un proyecto.

```
1 • USE empresa;  
2 • SELECT ROUND(AVG(Horas),0) 'Promedio Horas'  
3   FROM trabaja_en
```



The screenshot shows a database query result interface. At the top, there is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar, a table displays the query results. The table has one column labeled 'Promedio Horas' and one row with the value '18'.

Promedio Horas
18

SQL - NOW

- La función NOW() obtiene la fecha y hora actual.
- Es posible definir la precisión.
- Similar a las funciones CURRENT_TIMESTAMP y LOCALTIMESTAMP
- Ejemplo: Mostrar la fecha y hora actual.

SQL - NOW

```
1 • USE empresa;  
2 • SELECT NOW(),  
3   CURRENT_TIMESTAMP(3),  
4   CURRENT_TIMESTAMP,  
5   LOCALTIMESTAMP(6),  
6   LOCALTIMESTAMP  
7
```

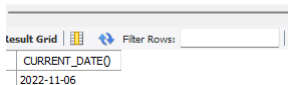
Result Grid				
Filter Rows: <input type="text"/>				
Export: <input type="button" value=""/>				
Wrap Cell Content: <input type="button" value=""/>				
NOW()	CURRENT_TIMESTAMP(3)	CURRENT_TIMESTAMP	LOCALTIMESTAMP(6)	LOCALTIMESTAMP
2022-11-06 13:12:54	2022-11-06 13:12:54.333	2022-11-06 13:12:54	2022-11-06 13:12:54.333864	2022-11-06 13:12:54

SQL - CURRENT_DATE

- La función `CURRENT_DATE()` obtiene la fecha actual.
- Es idéntica a la función `CURDATE()`
- Ejemplo: Mostrar la fecha actual.

SQL - CURRENT_DATE

- 1 • `USE empresa;`
- 2 • `SELECT CURRENT_DATE();`
- 3
- 4



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. Below the header, there is a table with one column named 'CURRENT_DATE()' and one row containing the date '2022-11-06'.

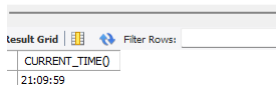
CURRENT_DATE()
2022-11-06

SQL - CURRENT_TIME

- La función `CURRENT_TIME()` obtiene la hora actual.
- Es idéntica a la función `CURTIME()`
- Ejemplo: Mostrar la hora actual.

SQL - CURRENT_TIME

```
1 • USE empresa;  
2 • SELECT CURRENT_TIME()  
3  
4
```







result Grid	
Filter Rows:	
CURRENT_TIME()	
21:09:59	

SQL - ADDDATE

- La función `ADDDATE()` adiciona o disminuye tiempo (definido en unidades de tiempo) a un fecha y retorna una fecha.
- `ADDDATE(date, INTERVAL value addunit)`
- `addunit`: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, etc.
- Ejemplo: Adicione dos días a la fecha "2022-10-22"

SQL - ADDDATE

```
1 • USE empresa;  
2 • SELECT  
3   ADDDATE("2022-10-22", INTERVAL 2 DAY) AS 'Fecha resultante'  
4
```



result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
Fecha resultante					
2022-10-24					

SQL - ADDTIME

- La función ADDTIME() adiciona o disminuye tiempo (definido en formato fecha y hora) a un fecha y retorna una fecha.
- ADDTIME(datetime, addtime)
- Ejemplo: Adicione dos días, 2 hours, 10 minutes, 5 seconds, and 3 microseconds a la fecha "2022-10-22 10:11:23.000000"

SQL - ADDTIME

- 1 • **USE** empresa;
- 2 • **SELECT**
- 3 **ADDTIME("2022-10-22 10:11:23.000000", "2:10:5.000003") AS 'Fecha resultante'**

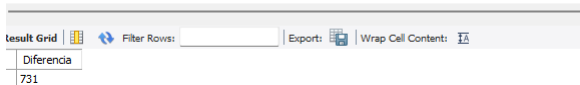
result Grid   Filter Rows: <input type="text"/>		Export: 	Wrap Cell Content: 
Fecha resultante			
2022-10-22 12:21:28.000003			

SQL - DATEDIFF

- La función DATEDIFF() retorna la diferencia de días entre dos fechas.
- DATEDIFF(date1, date2)
- Ejemplo: Muestre la diferencia de días entre las fechas "2020-01-06" y "2022-01-06"

SQL - DATEDIFF

```
1 • USE empresa;  
2 • SELECT  
3   DATEDIFF("2022-01-06", "2020-01-06") AS 'Diferencia'
```







Diferencia
731

SQL - DATE_FORMAT

- La función DATE_FORMAT() retorna una fecha en un formato específico. Tarea.
- DATE_FORMAT(date, format)
- Ejemplo: Muestre el mes en texto, el día y el año de la fecha "2020-01-06"

SQL - DATE_FORMAT

- 1 • **USE** empresa;
- 2 • **SELECT**
- 3 DATE_FORMAT("2022-01-06", "%M %d %Y") AS 'Fecha'

result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
Fecha					
January 06 2022					

SQL - EXTRACT

- La función `EXTRACT()` extrae parte de la fecha dada. Tarea
- `EXTRACT(part FROM date)`
- `part`: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, etc.
- Ejemplo: Extrae los minutos de la fecha "2022-10-22 10:35:22"

SQL - EXTRACT

- 1 • **USE** empresa;
- 2 • **SELECT**
- 3 **EXTRACT(MINUTE FROM "2022-01-06 10:35:22") AS 'Minutos'**

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Minutos				
35				

SQL - CASE

- La cláusula CASE analiza las condiciones y retorna el valor asignado cuando se cumple una condición.
- Ejemplo: Muestre el código, nombres, primer apellido y género (masculino o femenino) de cada empleado.

SQL - CASE

```
1 • USE empresa;  
2 SELECT CodEmpleado, Nombres, PrimerApellido,  
3 (CASE  
4     WHEN Sexo = 'M' THEN "Masculino"  
5     WHEN Sexo = 'F' THEN "Femenino"  
6     ELSE "No definido"  
7 END) AS 'Género'  
8 FROM empleado
```

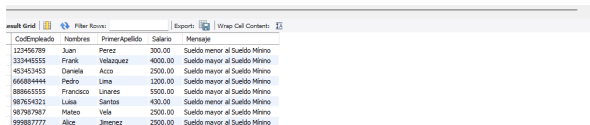
result Grid			
Filter Rows:		Export:	Wrap Cell Content:
CodEmpleado	Nombres	PrimerApellido	Género
123456789	Juan	Perez	Masculino
333445555	Frank	Velazquez	Masculino
453453453	Daniela	Acco	Femenino
666884444	Pedro	Lima	Masculino
888665555	Francisco	Linares	Masculino
987654321	Luisa	Santos	Femenino
987987987	Mateo	Vela	Masculino
999887777	Alice	Jimenez	Femenino

SQL - IF

- La cláusula IF analiza una condición y retorna el valor asignado en caso de verdad o el valor asignado en caso sea falso.
- Ejemplo: Muestre el código, nombres, primer apellido, sueldo e indique si el empleado gana más que el sueldo mínimo (700 soles) para todos los empleados.

SQL - IF

```
1 • USE empresa;  
2 SELECT CodEmpleado, Nombres, PrimerApellido, Salario,  
3 IF(Salario > 750, "Sueldo mayor al Sueldo Mínimo", "Sueldo menor al Sueldo Mínimo") 'Mensaje'  
4 FROM empleado
```



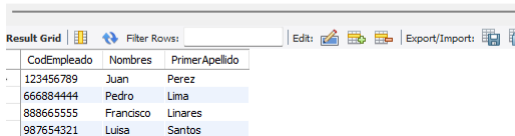
CodEmpleado	Nombres	PrimerApellido	Salario	Mensaje
123456789	Juan	Perez	300.00	Sueldo menor al Sueldo Mínimo
333445555	Frank	Velazquez	4000.00	Sueldo mayor al Sueldo Mínimo
453453453	Daniela	Acce	2500.00	Sueldo mayor al Sueldo Mínimo
666884444	Pedro	Lima	1200.00	Sueldo mayor al Sueldo Mínimo
888665555	Francisco	Linares	5500.00	Sueldo mayor al Sueldo Mínimo
987654321	Luisa	Santos	430.00	Sueldo menor al Sueldo Mínimo
987987987	Mateo	Vela	2500.00	Sueldo mayor al Sueldo Mínimo
999887777	Alice	Jimenez	2500.00	Sueldo mayor al Sueldo Mínimo

SQL - CAST

- La función CAST() convierte un valor (de cualquier tipo) a otro tipo. Tarea.
- CAST(value AS datatype)
- Ejemplo: Muestre el código, nombre y el pimer apellido de todos los empleados que tengan un número 6 en su código.

SQL - CAST

- 1 • `USE empresa;`
- 2 • `SELECT CodEmpleado, Nombres, PrimerApellido`
- 3 `FROM empleado`
- 4 `WHERE CAST(CodEmpleado AS CHAR(8)) LIKE '%6%'`



The screenshot shows a database application interface. At the top, there's a toolbar with icons for grid view, refresh, filter rows, edit, and export/import. Below the toolbar is a table with the following data:

CodEmpleado	Nombres	PrimerApellido
123456789	Juan	Perez
666884444	Pedro	Lima
888665555	Francisco	Linares
987654321	Luisa	Santos

SQL - Consideraciones finales

- Podemos utilizar las funciones y cláusula aprendidas combinadas entre si o con los JOINS. Tarea.
- Ejemplo: Muestre el código, nombre, primer apellido y cantidad de proyectos donde trabaja cada empleado.

SQL - Ejemplo - COUNT y INNER JOIN

- 1 • **USE** empresa;
- 2 • **SELECT** e.CodEmpleado, e.Nombres, e.PrimerApellido,
- 3 **COUNT**(te.CodProyecto) '**Cantidad de Proyectos**'
- 4 **FROM** empleado e
- 5 **INNER JOIN** trabaja_en te **ON** e.CodEmpleado = te.CodEmpleado
- 6 **GROUP BY** e.CodEmpleado, e.Nombres, e.PrimerApellido

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
CodEmpleado	Nombres	PrimerApellido	Cantidad de Proyectos			
123456789	Juan	Perez	2			
333445555	Frank	Velazquez	4			
453453453	Daniela	Acco	2			
666884444	Pedro	Lima	1			
888665555	Francisco	Linares	1			
987654321	Luisa	Santos	2			
987987987	Mateo	Vela	2			
999887777	Alice	Jimenez	2			