



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL
FORMATO ENTREGA DE EVIDENCIAS
CONSULTAS DML A LA BASE DE DATOS: INSERT, UPDATE, DELETE, SELECT

Presentado a: Cesar Marino Cuellar Chacón

Por Aprendiziz: David Santiago viveros Mera

Aprendices:

Ficha: 2874057

Competencia: construcción de software

Resultado: construir la base de datos para el software a partir del modelo de datos

Tecnólogo en Análisis y Desarrollo de Software
Servicio Nacional de Aprendizaje SENA
Centro de Teleinformática y Producción Industrial
Regional Cauca

Popayán, día 25 de noviembre del año 2024



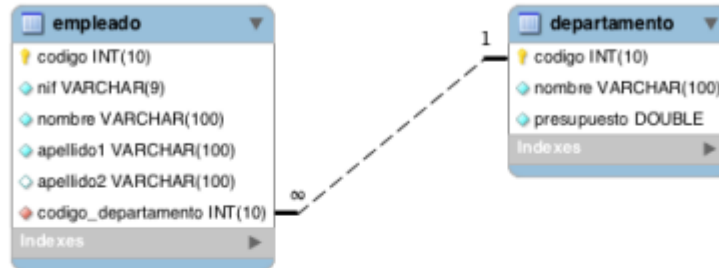
**PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL
FORMATO ENTREGA DE EVIDENCIAS**



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Taller Recuperación Bases de Datos

1. Realizar consultas a la siguiente base de Datos:



En el siguiente enlace encuentran el código sql de la base de datos que incluye su creación y unos datos para cada tabla.

<https://josejuansanchez.org/bd/ejercicios-consultas-sql/index.html>

Agregar una nueva tabla denominada **jefe_departamento** con los siguientes campos:

Campo	Tipo de Dato	PK	FK
Id_empleado	Int unsigned	X	X
Id_departamento	Int unsigned	X	X

Realizar las consultas que tiene el enlace de la siguiente manera:

- De las consultas de una tabla, realizar las múltiplos de 3: 1, 3, 6, 9, 12..., 36
- De las consultas multitable, composición interna, realizar todas las consultas impares: 1, 3, ..., 9
- De las consultas multitable, composición externa, realizar todas las consultas impares: 1, 3, 5
- De las consultas resumen, todas las consultas pares, 2, 4, 6, ..., 12
- De las subconsultas todas las subconsultas.



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
2 -- De las consultas de una tabla, realizar los múltiplos de 3: 1, 3, 6, 9,12.,36
3 -- 1 Lista el primer apellido de todos los empleados:
4 • SELECT apellido1 FROM empleado;
```

id	nif	nombre	apellido1	apellido2	id_departamento
1	32481596F	Aarón	Rivero	Gómez	1
2	Y5575632D	Adela	Salas	Díaz	2
3	R6970642B	Adolfo	Rubio	Flores	3
4	77705545E	Adrián	Suárez	NULL	4
5	17087203C	Marcos	Loyola	Méndez	5
6	38382980M	María	Santana	Moreno	1
7	80576669X	Pilar	Ruiz	NULL	2
8	71651431Z	Pepe	Ruiz	Santana	3
9	56399183D	Juan	Gómez	López	2
10	46384486H	Diego	Flores	Salas	5
11	67389283A	Marta	Herrera	Gil	1
12	41234836R	Irene	Salas	Flores	NULL
13	82635162B	Juan A...	Sáez	Guerrero	NULL
NULL	NULL	NULL	NULL	NULL	NULL

```
7 -- 3 Lista todas las columnas de la tabla empleado:
8 • SELECT * FROM empleado;
```

id	nif	nombre	apellido1	apellido2	id_departamento
1	32481596F	Aarón	Rivero	Gómez	1
2	Y5575632D	Adela	Salas	Díaz	2
3	R6970642B	Adolfo	Rubio	Flores	3
4	77705545E	Adrián	Suárez	NULL	4
5	17087203C	Marcos	Loyola	Méndez	5
6	38382980M	María	Santana	Moreno	1
7	80576669X	Pilar	Ruiz	NULL	2
8	71651431Z	Pepe	Ruiz	Santana	3
9	56399183D	Juan	Gómez	López	2
10	46384486H	Diego	Flores	Salas	5
11	67389283A	Marta	Herrera	Gil	1
12	41234836R	Irene	Salas	Flores	NULL
13	82635162B	Juan A...	Sáez	Guerrero	NULL
NULL	NULL	NULL	NULL	NULL	NULL

```
10 -- 6Lista el identificador de los departamentos de los empleados que aparecen en la tabla empleado,
11 -- eliminando los identificadores que aparecen repetidos:
```

```
12 • SELECT DISTINCT id_departamento FROM empleado;
```

```
13
```

id_departamento
NULL
1
2
3
4
5



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
16 -- 9 Lista el nombre y apellidos de los empleados en una única columna, convirtiendo todos los caracteres en minúscula:
17 • SELECT LOWER(CONCAT(nombre, ' ', apellido1, ' ', IFNULL(apellido2, ''))) AS nombre_completo
18 FROM empleado;
19
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

nombre_completo
aarón rivero gómez
adela salas díaz
adolfo rubio flores
adrián suárez
marcos loyola méndez
maría santana moreno
pilar ruiz
pepe ruiz santana
juan gómez lópez
diego flores salas
marta herrera gil
irene salas flores
juan antonio sáez gu...

```
21 -- 12 Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente:
22 • SELECT nombre, presupuesto
23 FROM departamento
24 ORDER BY presupuesto ASC;
25
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

nombre	presupuesto
Proyectos	0
Publicidad	0
Contabilidad	110000
Desarrollo	120000
Sistemas	150000
Gerencia	150000
Recursos Humanos	280000
I+D	375000

```
26 -- 15 Lista los apellidos y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta en primer lugar sus apellidos y luego su nombre:
27 • SELECT apellido1, apellido2, nombre
28 FROM empleado
29 ORDER BY apellido1, apellido2, nombre;
30
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

apellido1	apellido2	nombre
Flores	Salas	Diego
Gómez	López	Juan
Herrera	Gil	Marta
Loyola	Méndez	Marcos
Rivero	Gómez	Aarón
Rubio	Flores	Adolfo
Ruiz		Pilar
Ruiz	Santana	Pepe
Sáez	Guerrero	Juan Antonio
Salas	Díaz	Adela
Salas	Flores	Irene
Santana	Moreno	María
Suárez		Adrián



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

33 -- 18 Devuelve una lista con el nombre y el gasto, de los 2 departamentos que tienen mayor gasto:

```
34 • SELECT nombre, gastos
35 FROM departamento
36 ORDER BY gastos DESC
37 LIMIT 2;
```

38

39 -- 21 Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto mayor o igual a 150000 euros:

nombre	gastos
I+D	380000
Gerencia	70000

39 -- 21 Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto mayor o igual a 150000 euros:

```
40 • SELECT nombre, presupuesto
41 FROM departamento
42 WHERE presupuesto >= 150000;
```

43

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

nombre	presupuesto
Sistemas	150000
Recursos Humanos	280000
I+D	375000
Gerencia	150000

46 -- 24 Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre 100000 y 200000 euros. Sin utilizar el operador BETWEEN:

```
47 • SELECT nombre
48 FROM departamento
49 WHERE presupuesto < 100000 OR presupuesto > 200000;
```

50

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

nombre
Recursos Humanos
I+D
Proyectos
Publicidad

53 -- 27 Devuelve una lista con el nombre de los departamentos, gastos y presupuesto, de aquellos departamentos donde los gastos sean mayores que el presupuesto del que disponen:

```
54 • SELECT nombre, gastos, presupuesto
55 FROM departamento
56 WHERE gastos > presupuesto;
```

57

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

nombre	gastos	presupuesto
I+D	380000	375000
Publicidad	1000	0

59 -- 30 Lista todos los datos de los empleados cuyo segundo apellido sea NULL:

```
60 • SELECT *
61 FROM empleado
62 WHERE apellido2 IS NULL;
```

63

Result Grid | Filter Rows: | Edit: | Export/Import: | Fetch rows:

	id	nif	nombre	apellido1	apellido2	id_departamento
▶	4	77705545E	Adrián	Suárez	NULL	4
	7	80576669X	Pilar	Ruiz	NULL	2
*	NULL	NULL	NULL	NULL	NULL	NULL



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
64 -- 33 Lista todos los datos de los empleados cuyo segundo apellido sea Díaz o Moreno. Sin utilizar el operador IN:
65 • SELECT *
66 FROM empleado
67 WHERE apellido2 = 'Díaz' OR apellido2 = 'Moreno';
68
```

Result Grid						
Filter Rows:						
Edit:						
Export/Import:						
Wrap Cell Content:						
	id	nif	nombre	apellido1	apellido2	id_departamento
▶	2	Y5575632D	Adela	Salas	Díaz	2
	6	38382980M	María	Santana	Moreno	1
•	NULL	NULL	NULL	NULL	NULL	NULL

```
69 -- 36 Lista los nombres, apellidos y nif de los empleados que trabajan en los departamentos 2, 4 o 5:
70 • SELECT nombre, apellido1, apellido2, nif
71 FROM empleado
72 WHERE id_departamento = 2 OR id_departamento = 4 OR id_departamento = 5;
73
```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
	nombre	apellido1	apellido2	nif
▶	Adela	Salas	Díaz	Y5575632D
	Pilar	Ruiz	NULL	80576669X
	Juan	Gómez	López	56399183D
	Adrián	Suárez	NULL	77705545E
	Marcos	Loyola	Méndez	17087203C
	Diego	Flores	Salas	46384486H

```
75 -- De las consultas multitabla, composición interna, realizar todas las consultas impares: 1,3,...,9
76 -- 1. Listado con los empleados y los datos de los departamentos donde trabajan
77 • SELECT
78     e.id AS empleado_id,
79     e.nombre AS empleado_nombre,
80     e.apellido1,
81     e.apellido2,
82     d.id AS departamento_id,
83     d.nombre AS departamento_nombre,
84     d.presupuesto,
85     d.gastos
86 FROM
87     empleado e
88 LEFT JOIN
89     departamento d
90 ON
91     e.id_departamento = d.id;
```

Result Grid								
Filter Rows:								
Export:								
Wrap Cell Content:								
	empleado_id	empleado_nombre	apellido1	apellido2	departamento_id	departamento_nombre	presupuesto	gastos
▶	1	Aarón	Rivero	Gómez	1	Desarrollo	120000	6000
	2	Adela	Salas	Díaz	2	Sistemas	150000	21000
	3	Adolfo	Rubio	Flores	3	Recursos Humanos	280000	25000
	4	Adrián	Suárez	NULL	4	Contabilidad	110000	3000
	5	Marcos	Loyola	Méndez	5	I+D	375000	380000
	6	María	Santana	Moreno	1	Desarrollo	120000	6000
	7	Pilar	Ruiz	NULL	2	Sistemas	150000	21000
	8	Pepe	Ruiz	Santana	3	Recursos Humanos	280000	25000
	9	Juan	Gómez	López	2	Sistemas	150000	21000
	10	Diego	Flores	Salas	5	I+D	375000	380000
	11	Marta	Herrera	Gil	1	Desarrollo	120000	6000
	12	Irene	Salas	Flores	NULL	NULL	NULL	NULL
	13	Juan Antonio	Sáez	Guerrero	NULL	NULL	NULL	NULL

```
93 -- 3 Listado de identificador y nombre del departamento con empleados
94 • SELECT DISTINCT
95     d.id AS departamento_id,
96     d.nombre AS departamento_nombre
97 FROM
98     departamento d
99 INNER JOIN
100     empleado e
101 ON
102     d.id = e.id_departamento;
103
```

Result Grid	
Filter Rows:	
Export:	
Wrap Cell Content:	
	departamento_id
▶	1
	2
	3
	4
	5



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
104 -- 5 Nombre del departamento donde trabaja el empleado con NIF 38382980M
105 • SELECT
106     d.nombre AS departamento_nombre
107 FROM
108     empleado e
109 INNER JOIN
110     departamento d
111 ON
112     e.id_departamento = d.id
113 WHERE
114     e.nif = '38382980M';
115
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	departamento_nombre
►	Desarrollo

```
116 -- 7 Listado de empleados que trabajan en el departamento de I+D, ordenados alfabéticamente
117 • SELECT
118     e.id,
119     e.nombre,
120     e.apellido1,
121     e.apellido2
122 FROM
123     empleado e
124 INNER JOIN
125     departamento d
126 ON
127     e.id_departamento = d.id
128 WHERE
129     d.nombre = 'I+D'
130 ORDER BY
131     e.nombre,
132     e.apellido1,
133     e.apellido2;
134
---
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	id	nombre	apellido1	apellido2
►	10	Diego	Flores	Salas
	5	Marcos	Loyola	Méndez

```
135 -- 9 Listado con el nombre de los empleados que están en departamentos cuyo presupuesto no está entre 100000 y 200000 euros
136 • SELECT
137     e.nombre,
138     e.apellido1,
139     e.apellido2
140 FROM
141     empleado e
142 INNER JOIN
143     departamento d
144 ON
145     e.id_departamento = d.id
146 WHERE
147     d.presupuesto NOT BETWEEN 100000 AND 200000;
---
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	nombre	apellido1	apellido2
►	Adolfo	Rubio	Flores
	Pepe	Ruiz	Santana
	Marcos	Loyola	Méndez
	Diego	Flores	Salas



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
150 -- De las consultas multitabla, composición externa, realizar todas las consultas impares: 1,3,5
151 -- 1 Listado con todos los empleados junto con los datos de los departamentos donde trabajan. Incluye empleados sin departamento asociado.
152 • SELECT
153     empleado.id AS id_empleado,
154     empleado.nombre AS nombre_empleado,
155     empleado.apellido1 AS apellido1_empleado,
156     empleado.apellido2 AS apellido2_empleado,
157     departamento.id AS id_departamento,
158     departamento.nombre AS nombre_departamento
159 FROM empleado
160 LEFT JOIN departamento
161 ON empleado.id_departamento = departamento.id;
162
```

Result Grid						
Filter Rows:						
Export: Wrap Cell Content: I A						
	id_empleado	nombre_empleado	apellido1_empleado	apellido2_empleado	id_departamento	nombre_departamento
▶	1	Aarón	Rivero	Gómez	1	Desarrollo
	2	Adela	Salas	Díaz	2	Sistemas
	3	Adolfo	Rubio	Flores	3	Recursos Humanos
	4	Adrián	Suárez	NULL	4	Contabilidad
	5	Marcos	Loyola	Méndez	5	I+D
	6	María	Santana	Moreno	1	Desarrollo
	7	Pilar	Ruiz	NULL	2	Sistemas
	8	Pepe	Ruiz	Santana	3	Recursos Humanos
	9	Juan	Gómez	López	2	Sistemas
	10	Diego	Flores	Salas	5	I+D
	11	Marta	Herrera	Gil	1	Desarrollo
	12	Irene	Salas	Flores	NULL	NULL
	13	Juan Antonio	Sáez	Guerrero	NULL	NULL

```
163 -- 3 Listado de departamentos que no tienen ningún empleado asociado.
164 • SELECT
165     departamento.id AS id_departamento,
166     departamento.nombre AS nombre_departamento
167 FROM departamento
168 LEFT JOIN empleado
169 ON departamento.id = empleado.id_departamento
170 WHERE empleado.id IS NULL;
```

Result Grid		
Filter Rows:		
Export: Wrap Cell Content: I A		
	id_departamento	nombre_departamento
▶	6	Proyectos
	7	Publicidad
	8	Gerencia

```
172 -- 5 Listado con empleados sin departamento asociado y departamentos sin empleados asociados, ordenado por el nombre del departamento.
173 • SELECT
174     empleado.id AS id_empleado,
175     empleado.nombre AS nombre_empleado,
176     empleado.apellido1 AS apellido1_empleado,
177     empleado.apellido2 AS apellido2_empleado,
178     departamento.id AS id_departamento,
179     departamento.nombre AS nombre_departamento
180 FROM empleado
181 RIGHT JOIN departamento
182 ON empleado.id_departamento = departamento.id
183 WHERE empleado.id_departamento IS NULL OR departamento.id NOT IN (
184     SELECT id_departamento FROM empleado WHERE id_departamento IS NOT NULL
185 )
186 ORDER BY departamento.nombre;
```

Result Grid						
Filter Rows:						
Export: Wrap Cell Content: I A						
	id_empleado	nombre_empleado	apellido1_empleado	apellido2_empleado	id_departamento	nombre_departamento
▶	NULL	NULL	NULL	NULL	8	Gerencia
	NULL	NULL	NULL	NULL	6	Proyectos
	NULL	NULL	NULL	NULL	7	Publicidad



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
190 -- De las consultas resumen, todas las consultas pares, 2,4,6,...,12
191 -- 2 Media del presupuesto de todos los departamentos
192 • SELECT
193     AVG(presupuesto) AS media_presupuesto
194 FROM departamento;
195
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	media_presupuesto			
▶	148125			

```
196 -- 4 Nombre y presupuesto del departamento con menor presupuesto
197 • SELECT
198     nombre AS nombre_departamento,
199     presupuesto
200 FROM departamento
201 ORDER BY presupuesto ASC
202 LIMIT 1;
203
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	nombre_departamento	presupuesto		
▶	Proyectos	0		

```
204 -- 6 Nombre y presupuesto del departamento con mayor presupuesto
205 • SELECT
206     nombre AS nombre_departamento,
207     presupuesto
208 FROM departamento
209 ORDER BY presupuesto DESC
210 LIMIT 1;
211
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	nombre_departamento	presupuesto		
▶	I+D	375000		

```
212 -- 8 Número de empleados que no tienen NULL en su segundo apellido
213 • SELECT
214     COUNT(*) AS empleados_con_segundo_apellido
215 FROM empleado
216 WHERE apellido2 IS NOT NULL;
217
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	empleados_con_segundo_apellido			
▶	11			



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
218 -- 10 Nombre de los departamentos que tienen más de 2 empleados y el número de empleados asignados
219 • SELECT
220     departamento.nombre AS nombre_departamento,
221     COUNT(empleado.id) AS numero_empleados
222 FROM departamento
223 LEFT JOIN empleado
224 ON departamento.id = empleado.id_departamento
225 GROUP BY departamento.id
226 HAVING COUNT(empleado.id) > 2;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre_departamento	numero_empleados		
Desarrollo	3		
Sistemas	3		

```
228 -- 12 Número de empleados que trabajan en departamentos con presupuesto mayor a 200,000 euros
229 • SELECT
230     departamento.nombre AS nombre_departamento,
231     COUNT(empleado.id) AS numero_empleados
232 FROM departamento
233 LEFT JOIN empleado
234 ON departamento.id = empleado.id_departamento
235 WHERE departamento.presupuesto > 200000
236 GROUP BY departamento.id;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre_departamento	numero_empleados		
Recursos Humanos	2		
I+D	2		

```
239 -- De las subconsultas todas las subconsultas.
240 -- 1 Listado con todos los empleados que pertenecen al departamento de Sistemas (sin usar INNER JOIN)
241 • SELECT
242     empleado.nombre AS nombre_empleado,
243     empleado.apellido1 AS apellido1_empleado,
244     empleado.apellido2 AS apellido2_empleado
245 FROM empleado
246 WHERE empleado.id_departamento = (SELECT id FROM departamento WHERE nombre = 'Sistemas');
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre_empleado	apellido1_empleado	apellido2_empleado	
Adela	Salas	Díaz	
Pilar	Ruiz	NULL	
Juan	Gómez	López	



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
248      -- 2 Nombre del departamento con mayor presupuesto y la cantidad que tiene asignada
249  ●    SELECT
250          nombre AS nombre_departamento,
251          presupuesto
252  FROM departamento
253  WHERE presupuesto = (SELECT MAX(presupuesto) FROM departamento);
254
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	nombre_departamento	presupuesto		
▶	I+D	375000		

```
255      -- 3 Nombre del departamento con menor presupuesto y la cantidad que tiene asignada
256  ●    SELECT
257          nombre AS nombre_departamento,
258          presupuesto
259  FROM departamento
260  WHERE presupuesto = (SELECT MIN(presupuesto) FROM departamento);
261
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	nombre_departamento	presupuesto		
▶	I+D	375000		



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Consultas de insert, update, delete, select

- Insertar un departamento nuevo denominado Gerencia con un presupuesto de 150000 y gastos de 70000
- Actualizar el departamento de la Empleada Irene Salas Flores para que pertenezca ahora al departamento de la **Gerencia**.
- Actualizar el departamento del empleado Juan Antonio Sáez Guerrero para que pertenezca ahora al departamento de **Contabilidad**.
- Insertar a la tabla nueva **jefe_departamento** 3 registros, donde se evidencie la asignación de un jefe a los departamentos de la Gerencia, Desarrollo y Sistemas. Los jefes asignados deben pertenecer al departamento.
- Consulta donde muestre el nombre de empleado en un solo campo que incluya nombre, apellido1, apellido2, y también el nombre del departamento para aquellos empleados que son jefes de departamento.
- Eliminar los empleados que no tienen segundo apellido.

Procedimientos Almacenados, funciones y Triggers

- Crear un procedimiento almacenado denominado **cantidad_empleados_departamento** que reciba como parámetro de entrada el nombre del departamento y devuelva la cantidad de empleados del departamento en un parámetro de salida.
- Crear una función denominada **gastos_departamento_empleado** que reciba como parámetro el id del empleado y retorne los gastos del departamento al que pertenece el empleado.
- Crear un trigger denominado **check_presupuesto_gastos_departamento**
 - se ejecuta sobre la tabla departamento.
 - Se ejecuta antes de una operación de actualización.
 - Si el nuevo valor del presupuesto que se quiere insertar es negativo, se guarda como 0.
 - Si el nuevo valor de los gastos que se quiere insertar es superior a 100000, se debe guardar como 100000.



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
265 -- Consultas de insert, update, delete, select
266 -- Insertar un departamento nuevo denominado Gerencia con un presupuesto de 150000 y gastos de 70000
267 • INSERT INTO departamento (nombre, presupuesto, gastos)
268   VALUES ('Gerencia', 150000, 70000);
269
270 • SELECT * FROM departamento WHERE nombre = 'Gerencia';
```

Result Grid				
Filter Rows:				
Edit: Export/Import: Wrap Cell Content:				
	id	nombre	presupuesto	gastos
▶	8	Gerencia	150000	70000
	9	Gerencia	150000	70000
•	NULL	NULL	NULL	NULL

```
273 -- Actualizar el departamento de la Empleada Irene Salas Flores para que pertenezca ahora al departamento de la Gerencia.
274 • UPDATE empleado
275   SET id_departamento = (SELECT id FROM departamento WHERE nombre = 'Gerencia' LIMIT 1)
276   WHERE nif = '41234836R';
277
278 • SELECT nombre, id_departamento FROM empleado WHERE nif IN ('41234836R', '82635162B');
279
```

Result Grid		
Filter Rows:		
Export: Wrap Cell Content:		
	nombre	id_departamento
▶	Irene	8
	Juan Antonio	NULL

```
281 -- Actualizar el departamento del empleado Juan Antonio Sáez Guerrero para que pertenezca ahora al departamento de Contabilidad.
282 • UPDATE empleado
283   SET id_departamento = (SELECT id FROM departamento WHERE nombre = 'Contabilidad')
284   WHERE nif = '82635162B';
285 • SELECT nombre, id_departamento FROM empleado WHERE nif = '82635162B';
286
```

Result Grid		
Filter Rows:		
Export: Wrap Cell Content:		
	nombre	id_departamento
▶	Juan Antonio	4

```
291 -- Insertar a la tabla nueva jefe_departamento 3 registros, donde se evidencie la asignación de un jefe a los departamentos
292 -- de la Gerencia, Desarrollo y Sistemas. Los jefes asignados deben pertenecer al departamento.
293 -- jefe de departamento de Gerencia
294 • INSERT INTO jefe_departamento (id_empleado, id_departamento)
295   VALUES (
296     (SELECT id FROM empleado WHERE nif = '41234836R' LIMIT 1),
297     (SELECT id FROM departamento WHERE nombre = 'Gerencia' LIMIT 1)
298   );
299 -- jefe de departamento de Desarrollo
300 • INSERT INTO jefe_departamento (id_empleado, id_departamento)
301   VALUES (
302     (SELECT id FROM empleado WHERE nif = '32481596F' LIMIT 1),
303     (SELECT id FROM departamento WHERE nombre = 'Desarrollo' LIMIT 1)
304   );
305 -- jefe de departamento de Sistemas
306 • INSERT INTO jefe_departamento (id_empleado, id_departamento)
307   VALUES (
308     (SELECT id FROM empleado WHERE nif = 'Y5575632D' LIMIT 1),
309     (SELECT id FROM departamento WHERE nombre = 'Sistemas' LIMIT 1)
310   );
311 -- consulta
312 • SELECT CONCAT(e.nombre, ' ', e.apellido1, ' ', e.apellido2) AS nombre_empleado, d.nombre AS departamento
313   FROM empleado e
314   JOIN jefe_departamento jd ON e.id = jd.id_empleado
315   JOIN departamento d ON jd.id_departamento = d.id;
```

Result Grid		
Filter Rows:		
Export: Wrap Cell Content:		
	nombre_empleado	departamento
▶	Aarón Rivero Gómez	Desarrollo
	Adela Salas Díaz	Sistemas
	Irene Salas Flores	Gerencia



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
319 -- Consulta donde muestre el nombre de empleado en un solo campo que incluya nombre, apellido1, apellido2,
320 -- y también el nombre del departamento para aquellos empleados que son jefes de departamento.
321 • SELECT
322     CONCAT(e.nombre, ' ', e.apellido1, ' ', COALESCE(e.apellido2, '')) AS empleado_nombre,
323     d.nombre AS departamento
324 FROM empleado e
325 JOIN jefe_departamento jd ON e.id = jd.id_empleado
326 JOIN departamento d ON jd.id_departamento = d.id;
```

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

	empleado_nombre	departamento
▶	Aarón Rivero Gómez	Desarrollo
	Adela Salas Díaz	Sistemas
	Irene Salas Flores	Gerencia

```
328 -- Eliminar los empleados que no tienen segundo apellido.
329 • DELETE FROM empleado
330 WHERE apellido2 IS NULL;
```

Output

Action Output

	#	Time	Action
✓	114	18:07:05	DELETE FROM empleado WHERE apellido2 IS NULL

```
336 -- Procedimientos Almacenados, funciones y Triggers
337 -- Crear un procedimiento almacenado denominado cantidad_empleados_departamento que reciba como parámetro de
338 -- entrada el nombre del departamento y devuelva la cantidad de empleados del departamento en un parámetro de salida.
339 DELIMITER $$
340
341 • CREATE PROCEDURE cantidad_empleados_departamento(IN dept_nombre VARCHAR(100), OUT cantidad INT)
342 BEGIN
343     SELECT COUNT(*) INTO cantidad
344     FROM empleado e
345     JOIN departamento d ON e.id_departamento = d.id
346     WHERE d.nombre = dept_nombre;
347 END $$
348
349 DELIMITER ;
350
351 • CALL cantidad_empleados_departamento('Proyectos', @cantidad);
352 • SELECT @cantidad;
```

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

	@cantidad
▶	0



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

FORMATO ENTREGA DE EVIDENCIAS

```
355 -- Crear una función denominada gastos_departamento_empleado que reciba como parámetro el id del empleado y
356 -- retorne los gastos del departamento al que pertenece el empleado.
357 DELIMITER $$
358 • CREATE FUNCTION gastos_departamento_empleado(emp_id INT)
359 RETURNS DOUBLE
360 DETERMINISTIC
361 BEGIN
362     DECLARE dept_gastos DOUBLE;
363
364     SELECT d.gastos INTO dept_gastos
365     FROM empleado e
366     JOIN departamento d ON e.id_departamento = d.id
367     WHERE e.id = emp_id;
368
369     RETURN dept_gastos;
370 END $$
371 DELIMITER ;
372 • SELECT gastos_departamento_empleado(1);
```

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

gastos_departamento_empleado(1)
6000

```
376 -- Crear un trigger denominado check_presupuesto_gastos_departamento
377 -- o se ejecuta sobre la tabla departamento.
378 -- o Se ejecuta antes de una operación de actualización.
379 -- o Si el nuevo valor del presupuesto que se quiere insertar es negativo, se guarda como 0.
380 -- o Si el nuevo valor de los gastos que se quiere insertar es superior a 100000, se debe guardar como 100000.
381 DELIMITER $$
382
383 • CREATE TRIGGER check_presupuesto_gastos_departamento
384 BEFORE UPDATE ON departamento
385 FOR EACH ROW
386 BEGIN
387     -- Si el nuevo presupuesto es negativo, lo establecemos a 0
388     IF NEW.presupuesto < 0 THEN
389         SET NEW.presupuesto = 0;
390     END IF;
391
392     -- Si los nuevos gastos superan los 100000, los ajustamos a 100000
393     IF NEW.gastos > 100000 THEN
394         SET NEW.gastos = 100000;
395     END IF;
396 END $$
397
398 DELIMITER ;
399
```

123 10:12:51 CREATE TRIGGER check_presupuesto_gastos_departamento BEFORE UPDATE ON departamento FOR EACH ROW BEGIN - ... 0 row(s) affected 0.079 sec

- Para probar el trigger anterior realizar las siguientes actualizaciones:
 - Actualizar el presupuesto del departamento **Proyectos** con un valor de -120000 y gastos de 250000
 - Actualizar el presupuesto del departamento de **Publicidad** con un valor de 250000 y gastos de 100000



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
400 • -- Para probar el trigger anterior realizar las siguientes actualizaciones:
401 -- o Actualizar el presupuesto del departamento Proyectos con un valor de -120000 y gastos de 250000
402 -- o Actualizar el presupuesto del departamento de Publicidad con un valor de 250000 y gastos de 100000
403 -- Actualizar el presupuesto del departamento 'Proyectos' con un valor negativo y gastos altos
404 UPDATE departamento
405 SET presupuesto = -120000, gastos = 250000
406 WHERE nombre = 'Proyectos';
407
408 • ALTER TABLE departamento MODIFY COLUMN presupuesto DOUBLE;
409 • SELECT * FROM departamento
410 WHERE nombre = 'Proyectos';
```

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

	id	nombre	presupuesto	gastos
▶	6	Proyectos	0	100000
★	NULL	NULL	NULL	NULL

```
415 -- Actualizar el presupuesto del departamento 'Publicidad' con un valor positivo y gastos altos
416 • UPDATE departamento
417 SET presupuesto = 250000, gastos = 100000
418 WHERE nombre = 'Publicidad';
419 • SELECT * FROM departamento
420 WHERE nombre = 'Publicidad';
```

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

	id	nombre	presupuesto	gastos
▶	7	Publicidad	250000	100000
★	NULL	NULL	NULL	NULL