

DEPARTAMENTO:	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN	CARRERA:	INGENIERÍA EN SOFTWARE		
ASIGNATURA:	SISTEMAS DE BASE DE DATOS	PERIODO LECTIVO:	MAYO – SEPTIEMBRE 2023	NIVEL:	4to
DOCENTE:	ING. ELEANA JEREZ, MSc.	NRC:	9752	PRÁCTICA N°:	2
TEMA DE LA PRÁCTICA:	Consultas en MySQL/PostgreSQL				

INTRODUCCIÓN:

En un entorno de base de datos relacional, las subconsultas son esenciales en herramientas como PostgreSQL, pgAdmin, MySQL Workbench y phpMyAdmin. Estas consultas anidadas permiten dividir problemas complejos en pasos más sencillos y lograr coherencia entre conjuntos de datos interconectados. Este estudio a detalle proporciona información relevante acerca de cómo las subconsultas recuperan, combinan y transforman datos de manera eficiente. Además, el presente tema brinda nuevos conocimientos y habilidades de gran impacto e importancia para las subconsultas tanto de principiantes como de expertos en PostgreSQL, pgAdmin, MySQL Workbench y phpMyAdmin. En aplicaciones sencillas como de mayor dificultad son necesarias aprender a generar subconsultas y componentes básicos de consultas, ya que abarcan un valor primordial en las bases de datos. Los ejemplos prácticos cubren búsquedas en varias tablas y cálculos en conjuntos de datos anidados.

OBJETIVOS:

Objetivo General

Este estudio tiene como objetivo principal proporcionar un profundo entendimiento sobre el uso efectivo de subconsultas, incluyendo ordenamiento y agrupaciones, en bases de datos relacionales. Se centra en entornos como PostgreSQL y MySQL Workbench buscando que los participantes adquieran habilidades sólidas para emplear subconsultas de manera estratégica y eficiente en la manipulación y extracción de datos interrelacionados.

Objetivos Específicos

- Explorar la naturaleza y función de las subconsultas en el contexto de las bases de datos relacionales, comprendiendo cómo dividen problemas complejos en componentes más manejables.
- Analizar las capacidades y características de las herramientas de administración de bases de datos mencionadas (PostgreSQL, pgAdmin, MySQL Workbench y phpMyAdmin) en relación con la creación y ejecución de subconsultas.
- Proporcionar ejemplos prácticos que involucren búsqueda, manipulación y análisis de datos en tablas interconectadas, utilizando subconsultas con ordenamiento y agrupaciones para resolver desafíos típicos en la gestión de bases de datos.



EQUIPOS:

- Jose: PC (Hp Omen, I5 de 7 generación, 8 GB de ram, Gtx 1050).
- Martin: PC (Asus Tuf F17, I5 de 11ava, 16 GB de ram, Gtx 1650).
- Ricardo: PC (Dell, I7 de 11ava generación, 8 GB ram).
- Angelo: PC (Lenovo, I5 de 11ava generación, 8GB de ram, GTX 1050)

HERRAMIENTAS USADAS:

- Postgres 15 (pgAdmin4)
- MySQL
- PhpMyAdmin
- CMD
- DBeaver
- Xampp

INSTRUCCIONES:

EJERCICIO 1

Realizar las siguientes consultas en MySQL

1. Actualizar el título de Perry Pepe a "system enginner"
2. Actualizar el desde el primero de enero de 2023 hasta el 31 de dicimebre de 2023 de Margareta Markovitch que corresponde al departamento 001
3. Borrar todos los empleados que pertenezcan al departamento de ventas y que sean hombres
4. Actualizar el nombre de del departamento de recursos humanos a español
5. Listas todos los empleados que ganan más de 40.000 dólares por contrato
6. Eliminar a los empleados que ganan menos de 40.000 dólares por contrato
7. Listar a todos los empleados que ganan lo mismo que Perry Pepe
8. Actualizar la fecha de contratación del empleado Perry Pepe a el 2023-01-01

EJERCICIO 2

Realizar las siguientes consultas en PostgreSQL

1. Ordenar a los clientes por su primer nombre (nombre, apellido) en orden ascendente.
2. Mostrar el primer nombre y apellido, mostrar los resultados en orden descendente por apellido.
3. Mostrar el nombre y el apellido, ordenar el nombre de forma ascendente y apellido de forma descendente.
4. Obtener el ID del cliente y agrupar por el ID del cliente. Mostrar y agrupar por el ID.
5. To get the total amount that each customer has been payed - Obtener el importe total que se ha pagado a cada cliente.
6. Mostrar nombre y apellido de los clientes en una sola columna y en otra columna el total pagado por los clientes; agrupar por nombre completo y ordenar por el total pagado en forma descendente.
7. To find the number of payment transactions that each staff has processed - Averiguar el número de transacciones que ha procesado cada empleado.
8. To show customer_id, staff_id and total amount. Group by staff_id customer_id. Order by customer_id.



ACTIVIDADES POR DESARROLLAR EN EL LABORATORIO:

EJERCICIO 1

Realizar las siguientes consultas en MySQL

1. Actualizar el título de Perry Pepe a "system enginner"

```
1 UPDATE titles
2 INNER JOIN employees ON employees.emp_no=titles.emp_no
3 SET titles.title='System Enginner'
4 WHERE employees.first_name='Perry' and employees.last_name='Pepe';
```

✓ 2 filas afectadas. (La consulta tardó 0.2234 segundos.)

```
UPDATE titles INNER JOIN employees ON employees.emp_no=titles.emp_no SET titles.title='System Enginner' WHERE employees.first_name='Perry' and employees.last_name='Pepe';
```

[Editar en línea] [Editar] [Crear código PHP]

```
1 • SELECT first_name, last_name, titles.title FROM employees
2 INNER JOIN titles ON titles.emp_no=employees.emp_no
3 WHERE employees.first_name='Perry' and employees.last_name='Pepe';
```

Result Grid		
Filter Rows: <input type="text"/>		
Export: Wrap Cell Content:		
first_name	last_name	title
Perry	Pepe	System Enginner
Perry	Pepe	System Enginner

- Actualizar el desde el primero de enero de 2023 hasta el 31 de dicimebre de 2023 de Margareta Marcovitch que corresponde al departamento 001

```
1 UPDATE dept_emp
2 INNER JOIN employees ON employees.emp_no=dept_emp.emp_no
3 SET dept_emp.from_date='2023-01-01' , dept_emp.to_date='2023-12-31'
4 WHERE employees.first_name='Margareta' and employees.last_name='Markovitch ';
```

✓ 1 fila afectada. (La consulta tardó 0,1541 segundos.)

UPDATE dept_emp INNER JOIN employees ON employees.emp_no=dept_emp.emp_no SET dept_emp.from_date='2023-01-01' , dept_emp.to_date='2023-12-31' WHERE employees.first_name='Margareta' and employees.last_name='Markovitch ';

[Editar en línea] [Editar] [Crear código PHP]

```
1 • SELECT employees.first_name,employees.last_name,from_date, to_date FROM dept_emp
2 INNER JOIN employees ON employees.emp_no=dept_emp.emp_no
3 WHERE employees.first_name='Margareta' and employees.last_name='Markovitch';
```

first_name	last_name	from_date	to_date
Margareta	Markovitch	2023-01-01	2023-12-31

- Borrar todos los empleados que pertenezcan al departamento de ventas y que sean hombres

```
1 • DELETE departments
2 FROM departments
3 INNER JOIN dept_emp ON dept_emp.dept_no=departments.dept_no
4 INNER JOIN employees ON dept_emp.emp_no=employees.emp_no
5 WHERE departments.dept_name='Sales' and employees.gender='M';
```

#	Time	Action	Message
✓ 29	18:28:56	SELECT employees.first_name,employees.last_name,from_date, to_date FROM dept_emp INNER JOIN employee...	1 row(s) returned
✓ 30	18:30:26	SELECT * FROM employees departments LIMIT 0, 1000	9 row(s) returned
✓ 31	18:34:52	SELECT * FROM employees departments LIMIT 0, 1000	9 row(s) returned
✓ 32	18:35:24	SELECT * FROM employees employees LIMIT 0, 1000	1000 row(s) returned
✓ 33	18:36:20	DELETE departments FROM departments INNER JOIN dept_emp ON dept_emp.dept_no=departments.dept_no ...	1 row(s) affected

4. Actualizar el nombre de del departamento de recursos humanos a español

```

1  UPDATE departments
2  SET dept_name='Recursos Humanos'
3  WHERE dept_name='Human Resources';

```

Output

#	Time	Action	Message
31	18:34:52	SELECT * FROM employees.departments LIMIT 0, 1000	9 row(s) returned
32	18:35:24	SELECT * FROM employees.employees LIMIT 0, 1000	1000 row(s) returned
33	18:36:20	DELETE departments FROM departments INNER JOIN dept_emp ON dept_emp.dept_no=departments.dept_no ...	1 row(s) affected
34	18:37:55	UPDATE departments SET dept_name='Recursos Humanos' WHERE dept_name='Human Resources'	Error Code: 1146. Table 'employees.departments' doesn't exist
35	18:38:08	UPDATE departments SET dept_name='Recursos Humanos' WHERE dept_name='Human Resources'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

```

1 • SELECT * FROM employees.departments WHERE dept_name='Recursos Humanos';

```

Result Grid

dept_no	dept_name
d003	Recursos Humanos
NULL	NULL



5. Listas todos los empleados que ganan más de 40.000 dólares por contrato

```
1 • select first_name, last_name, salaries.salary
2 FROM employees
3 INNER JOIN salaries ON salaries.emp_no=employees.emp_no
4 WHERE salaries.salary>40000;
```

Limit to 1000 rows

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

	first_name	last_name	salary
▶	Georgi	Facello	60117
	Georgi	Facello	62102
	Georgi	Facello	66074
	Georgi	Facello	66596
	Georgi	Facello	66961
	Georgi	Facello	71046

6. Eliminar a los empleados que ganan menos de 40.000 dólares por contrato

```
1 DELETE employees
2 FROM employees
3 JOIN salaries ON salaries.emp_no=employees.emp_no
4 WHERE salaries.salary < 40000;
5
```

9958 filas eliminadas. (La consulta tardó 4,6041 segundos.)

DELETE employees FROM employees JOIN salaries ON salaries.emp_no=employees.emp_no WHERE salaries.salary < 40000;

[Editar en línea] [Editar] [Crear código PHP]



```
1 • select first_name, last_name, salaries.salary
2 FROM employees
3 INNER JOIN salaries ON salaries.emp_no=employees.emp_no
4 WHERE salaries.salary<40000;
```

Limit to 1000 rows

Result Grid

first_name	last_name	salary
------------	-----------	--------

7. Listar a todos los empleados que ganan lo mismo que Perry Pepe

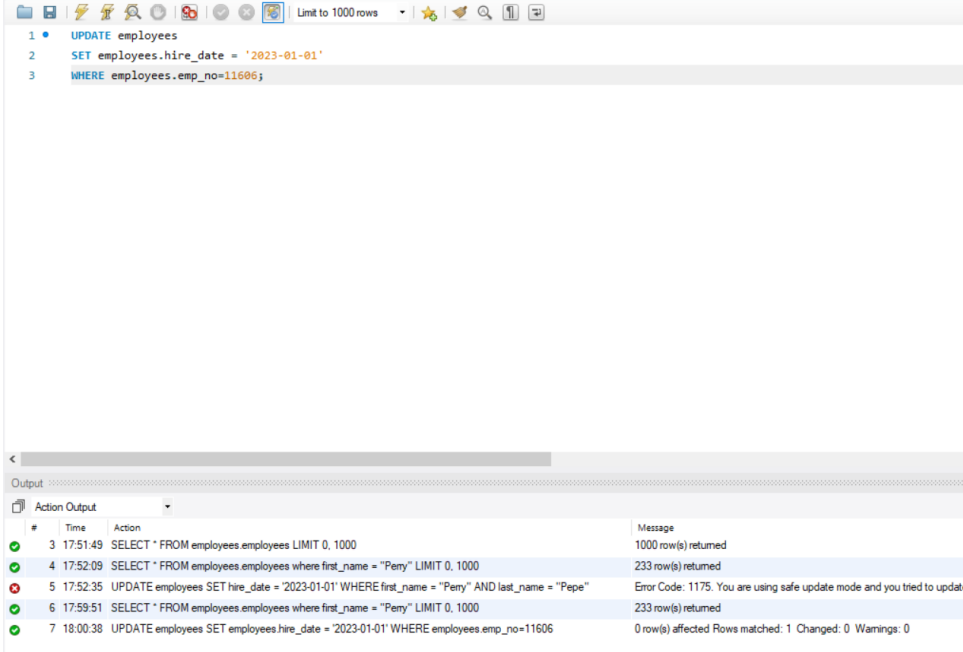
```
1 • SELECT first_name, last_name
2 FROM employees
3 INNER JOIN salaries ON salaries.emp_no = employees.emp_no
4 WHERE salaries.salary = (
5     SELECT salary
6     FROM salaries
7     INNER JOIN employees ON employees.emp_no = salaries.emp_no
8     WHERE employees.first_name = 'Perry' AND employees.last_name = 'Pepe'
9     LIMIT 1
10 );
```

Result Grid

first_name	last_name
Anneke	Preusig
Patricio	Bridgland
Eberhardt	Terkki
Guoxiang	Nooteboom
Mayuko	Warwick
Prasadram	Heyers
Karsten	Joslin
Huan	Lortz
Alejandro	Brender

Result 12 x

8. Actualizar la fecha de contratación del empleado Perry Pepe a el 2023-01-01



```

1 • UPDATE employees
2   SET employees.hire_date = '2023-01-01'
3   WHERE employees.emp_no=11606;

```

Output

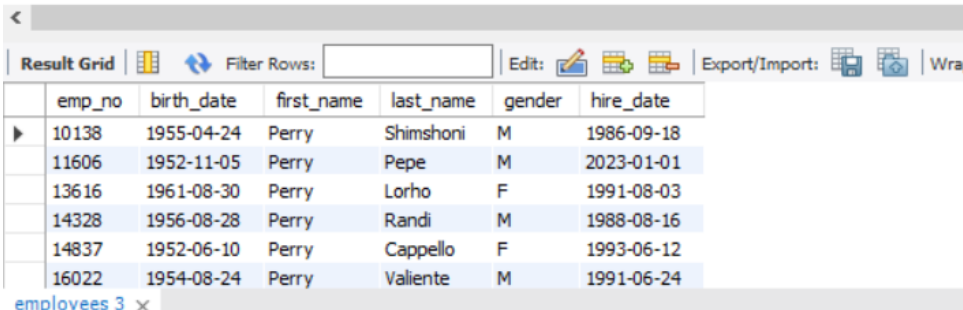
#	Time	Action	Message
3	17:51:49	SELECT * FROM employees.employees LIMIT 0, 1000	1000 row(s) returned
4	17:52:09	SELECT * FROM employees.employees where first_name = "Perry" LIMIT 0, 1000	233 row(s) returned
5	17:52:35	UPDATE employees SET hire_date = '2023-01-01' WHERE first_name = "Perry" AND last_name = "Pepe"	Error Code: 1175. You are using safe update mode and you tried to update
6	17:59:51	SELECT * FROM employees.employees where first_name = "Perry" LIMIT 0, 1000	233 row(s) returned
7	18:00:38	UPDATE employees SET employees.hire_date = '2023-01-01' WHERE employees.emp_no=11606	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0

```

1 • SELECT * FROM employees.employees where first_name = "Perry";

```

Execute the selected portion of the script or everything, if there is no selection



	emp_no	birth_date	first_name	last_name	gender	hire_date
▶	10138	1955-04-24	Perry	Shimshoni	M	1986-09-18
	11606	1952-11-05	Perry	Pepe	M	2023-01-01
	13616	1961-08-30	Perry	Lorho	F	1991-08-03
	14328	1956-08-28	Perry	Randi	M	1988-08-16
	14837	1952-06-10	Perry	Cappello	F	1993-06-12
	16022	1954-08-24	Perry	Valiente	M	1991-06-24







employees 3 x

EJERCICIO 2

Realizar las siguientes consultas en PostgreSQL

1. Ordenar a los clientes por su primer nombre (nombre, apellido) en orden ascendente.

Query		
1	SELECT first_name,last_name	
2	FROM customer	
3	ORDER BY first_name ASC;	

Data Output Messages Notifications		
     		
	first_name character varying (45) 🔒	last_name character varying (45) 🔒
1	Aaron	Selby
2	Adam	Gooch
3	Adrian	Clary
4	Agnes	Bishop
5	Alan	Kahn
6	Albert	Crouse
7	Alberto	Henning
8	Alex	Gresham
9	Alexander	Fennell
10	Alfred	Casillas
11	Alfredo	Mcadams
12	Alice	Stewart
13	Alicia	Mills
14	Allan	Cornish
Total rows: 599 of 599 Query complete 00:00:00.091		



Query

```
1 SELECT first_name,last_name
2 FROM customer
3 ORDER BY first_name ASC;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)
14	Allan	Cornish
15	Allen	Butterfield
16	Allison	Stanley
17	Alma	Austin
18	Alvin	Deloach
19	Amanda	Carter
20	Amber	Dixon
21	Amy	Lopez
22	Ana	Bradley
23	Andre	Rapp
24	Andrea	Henderson
25	Andrew	Purdy
26	Andy	Vanhorn
27	Angel	Barclay
Total rows: 599 of 599		Query complete 00:00:00.091



2. Mostrar el primer nombre y apellido, mostrar los resultados en orden descendente por apellido.

Query		
1	SELECT first_name,last_name	
2	FROM customer	
3	ORDER BY last_name DESC ;	

Data Output		
Messages		
Notifications		
	first_name character varying (45)	last_name character varying (45)
1	Cynthia	Young
2	Marvin	Yee
3	Luis	Yanez
4	Brian	Wyman
5	Brenda	Wright
6	Tyler	Wren
7	Florence	Woods
8	Lori	Wood
9	Virgil	Wofford
10	Darren	Windham
11	Susan	Wilson
12	Bernice	Willis
13	Gina	Williamson
14	Linda	Williams
Total rows: 599 of 599		
Query complete 00:00:00.112		



Query

```
1 SELECT first_name,last_name
2 FROM customer
3 ORDER BY last_name DESC;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)
14	Linda	Williams
15	Jon	Wiles
16	Roy	Whiting
17	Betty	White
18	Lucy	Wheeler
19	Fred	Wheat
20	Mitchell	Westmoreland
21	Edna	West
22	Sheila	Wells
23	Marlene	Welch
24	Ronald	Weiner
25	Ethel	Webb
26	Yolanda	Weaver
27	Mike	Way
Total rows: 599 of 599		Query complete 00:00:00.112



3. Mostrar el nombre y el apellido, ordenar el nombre de forma ascendente y apellido de forma descendente.

Query

```
1 SELECT first_name,last_name
2 FROM customer
3 ORDER BY last_name DESC,first_name ASC;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)
1	Cynthia	Young
2	Marvin	Yee
3	Luis	Yanez
4	Brian	Wyman
5	Brenda	Wright
6	Tyler	Wren
7	Florence	Woods
8	Lori	Wood
9	Virgil	Wofford
10	Darren	Windham
11	Susan	Wilson
12	Bernice	Willis
13	Gina	Williamson
14	Linda	Williams
Total rows: 599 of 599		Query complete 00:00:00.088



Query

```
1 SELECT first_name,last_name
2 FROM customer
3 ORDER BY last_name DESC,first_name ASC;
```

Data Output

Messages

Notifications



	first_name character varying (45)	last_name character varying (45)
14	Linda	Williams
15	Jon	Wiles
16	Roy	Whiting
17	Betty	White
18	Lucy	Wheeler
19	Fred	Wheat
20	Mitchell	Westmoreland
21	Edna	West
22	Sheila	Wells
23	Marlene	Welch
24	Ronald	Weiner
25	Ethel	Webb
26	Yolanda	Weaver
27	Mike	Way

Total rows: 599 of 599

Query complete 00:00:00.088



4. Obtener el ID del cliente y agrupar por el ID del cliente. Mostrar y agrupar por el ID.

Query

```
1 SELECT customer_id
2 FROM customer
3 GROUP BY customer_id;
4
```

Data Output Messages Notifications



	customer_id [PK] integer
1	184
2	87
3	477
4	273
5	550
6	394
7	51
8	272
9	70
10	190
11	350
12	539
13	554
14	278

Total rows: 599 of 599 Query complete 00:00:00.119

Query

```
1 SELECT customer_id
2 FROM customer
3 GROUP BY customer_id;
4
```

Data Output Messages Notifications



	customer_id [PK] integer
15	424
16	406
17	176
18	576
19	309
20	292
21	509
22	271
23	22
24	156
25	417
26	556
27	475
28	529

Total rows: 599 of 599 Query complete 00:00:00.119



5. To get the total amount that each customer has been paid - Obtener el importe total que se ha pagado a cada cliente.

Query

```
1 SELECT customer.customer_id, SUM(payment.amount) AS Pago_total
2 FROM customer
3 INNER JOIN payment ON payment.customer_id=customer.customer_id
4 GROUP BY customer.customer_id;
```

Data Output Messages Notifications



	customer_id [PK] integer	pago_total numeric
1	184	80.80
2	87	137.72
3	477	106.79
4	273	130.72
5	550	151.69
6	51	123.70
7	394	77.80
8	272	65.87
9	70	75.83
10	190	102.75
11	350	63.79
12	539	84.80
13	554	95.80
14	278	71.79

Total rows: 599 of 599 Query complete 00:00:00.083



Query

```
1 SELECT customer.customer_id, SUM(payment.amount) AS Pago_total
2 FROM customer
3 INNER JOIN payment ON payment.customer_id=customer.customer_id
4 GROUP BY customer.customer_id;
```

Data Output Messages Notifications

	customer_id [PK] integer	pago_total numeric
15	424	109.71
16	406	121.69
17	176	151.68
18	576	135.68
19	309	113.75
20	292	102.78
21	509	86.82
22	271	56.84
23	22	94.81
24	156	89.77
25	417	91.78
26	556	72.82
27	475	97.78
28	529	115.72

Total rows: 599 of 599 Query complete 00:00:00.083

6. Mostrar nombre y apellido de los clientes en una sola columna y en otra columna el total pagado por los clientes; agrupar por nombre completo y ordenar por el total pagado en forma descendente.

Query Query History

```
1 SELECT CONCAT (c.first_name,',',c.last_name) AS nombre_completo, SUM(p.amount) AS Pago_total
2 FROM customer c
3 JOIN payment p ON c.customer_id=p.customer_id
4 GROUP BY c.first_name,c.last_name
5 ORDER BY Pago_total DESC;
```

Data Output Messages Notifications

	nombre_completo text	pago_total numeric
1	EleanorHunt	211.55
2	KarlSeal	208.58
3	MarionSnyder	194.61
4	RhondaKennedy	191.62
5	ClaraShaw	189.60
6	TommyCollazo	183.63
7	AnaBradley	167.67
8	CurtisIrby	167.62
9	MarciaDean	166.61
10	MikeWay	162.67
11	ArnoldHavens	161.68
12	WesleyBull	158.65
13	GordonAllard	157.69

Total rows: 599 of 599 Query complete 00:00:00.066

7. To find the number of payment transactions that each staff has processed - Averiguar el número de transacciones que ha procesado cada empleado.

```
Query    Query History
1  SELECT s.staff_id, CONCAT (s.first_name,',',s.last_name) AS Nombre_Staff,
2  COUNT(p.payment_id) AS Transacciones
3  FROM staff s
4  JOIN payment p ON s.staff_id=p.staff_id
5  GROUP BY s.staff_id,s.first_name, s.last_name
6  ORDER BY Nombre_Staff:

Data Output    Messages    Notifications

[Icons: SQL, CSV, JSON, etc.]

staff_id [PK] integer | nombre_staff text | transacciones bigint
1 | 2 | JonStephens | 7304
2 | 1 | MikeHillyer | 7292
```



8. To show customer_id, staff_id and total amount. Group by staff_id customer_id. Order by customer_id.

Query

```
1 SELECT payment.customer_id, staff.staff_id, SUM(payment.amount) AS total_amount
2 FROM staff
3 INNER JOIN payment ON staff.staff_id=payment.staff_id
4 GROUP BY staff.staff_id, payment.customer_id
5 ORDER BY payment.customer_id;
6
```

Data Output Messages Notifications

	customer_id smallint	staff_id integer	total_amount numeric
1	1	2	53.85
2	1	1	60.85
3	2	2	67.88
4	2	1	55.86
5	3	1	59.88
6	3	2	70.88
7	4	2	31.90
8	4	1	49.88
9	5	1	63.86
10	5	2	70.79
11	6	1	53.85
12	6	2	30.90
13	7	1	69.84
14	7	2	60.88

Total rows: 1000 of 1198 Query complete 00:00:00.083

Query

```
1 SELECT payment.customer_id, staff.staff_id, SUM(payment.amount) AS total_amount
2 FROM staff
3 INNER JOIN payment ON staff.staff_id=payment.staff_id
4 GROUP BY staff.staff_id, payment.customer_id
5 ORDER BY payment.customer_id;
6
```

Data Output Messages Notifications

	customer_id smallint	staff_id integer	total_amount numeric
15	8	1	57.86
16	8	2	27.91
17	9	2	38.92
18	9	1	39.88
19	10	1	40.88
20	10	2	53.88
21	11	2	45.89
22	11	1	53.88
23	12	1	21.92
24	12	2	71.82
25	13	2	54.90
26	13	1	76.83
27	14	1	55.84
28	14	2	40.93

Total rows: 1000 of 1198 Query complete 00:00:00.083

RESULTADOS OBTENIDOS:

- Las consultas planteadas en el laboratorio resaltan la importancia de tener habilidades sólidas en SQL al interactuar con bases de datos. La capacidad de escribir consultas precisas y eficientes es esencial para realizar operaciones de actualización, eliminación, agrupación y ordenamiento de datos de manera efectiva. Sin una comprensión profunda de SQL, es fácil cometer errores o no obtener los resultados deseados.
- Los ejercicios planteados en el laboratorio ponen de relieve la necesidad de garantizar la integridad y la seguridad de los datos al realizar operaciones de actualización, eliminación y agrupación. Es esencial ser cauteloso al modificar o eliminar registros, especialmente en entornos de producción, para evitar pérdida de datos o corrupción. Además, asegurarse de que las consultas estén protegidas contra inyecciones SQL y de que los permisos de usuario sean adecuados es fundamental para mantener la seguridad de la base de datos.
- Los ejercicios se realizan en sistemas de gestión de bases de datos diferentes (MySQL y PostgreSQL) esto demuestra que las habilidades en SQL son transferibles, pero también resalta que cada sistema puede tener sus propias particularidades en términos de sintaxis y características específicas. La adaptabilidad entre sistemas es crucial para trabajar en diversos entornos y maximizar la eficiencia al interactuar con diferentes bases de datos.

CONCLUSIONES:

- Las subconsultas son esenciales en bases de datos relacionales. Ayudan a dividir problemas complejos en pasos más sencillos, mejorando la eficiencia y coherencia en la gestión de datos.
- Las herramientas como PostgreSQL, pgAdmin, MySQL Workbench y phpMyAdmin han sido evaluadas en su habilidad para crear y ejecutar subconsultas, brindando claridad sobre su potencial en la gestión de datos.
- La aplicación práctica de subconsultas con ordenamiento y agrupaciones muestra su valor para resolver problemas comunes de análisis y manipulación de datos, realzando la relevancia de estas técnicas en la toma de decisiones informadas y en la organización de resultados de manera útil.

RECOMENDACIONES:

- ✓ Para realizar las tablas resultantes de las consultas mediante SQL es conveniente empezar analizando desde las expresiones más pequeñas hasta las expresiones más grandes, de esta manera se consigue dibujar las tablas resultantes sin tantas complicaciones.
- ✓ Antes de comenzar a realizar las consultas y sus tablas resultantes se debe tener un previo conocimiento de las cláusulas principales de postgres y mysql (FROM y SELECT), ya que son los que se van a usar para dar solución a los enunciados de los ejercicios planteados.
- ✓ Para realizar las diferentes consultas acordes al enunciado de cada literal de cada ejercicio, se debe leer las veces que sean necesarias hasta entender el contexto que nos pide, es decir, revisar una y otra vez las tablas de las bases de datos y comprender de que tablas nos pide el enunciado extraer la información o los datos necesarios para construir la expresión en SQL y correctamente.

ELABORADO POR:

Rivadeneira Gómez Ricardo Xavier
Medina Armijos Martin Andrés
Yanacallo Monta Angelo Josué
Imbaquinga Guaña Jose Ricardo

ESTUDIANTE