

Informe de Subconsultas SQL

Presentado por:

Michael Alejandro Papamija Pantoja

Docente:

Brayan Arcos

Materia:

Desarrollo de Base de Datos

Instituto Tecnológico del Putumayo

Mocoa – Putumayo

2024

Tabla de contenido

I. Introducción	3
Contexto y Relevancia	3
Tipos de Subconsultas Analizadas	3
1. Subconsultas WHERE	4
1.1. Productos sin movimientos	4
1.2. Clientes con gastos mayor al promedio	4
2. Subconsultas FROM	5
2.1. Promedio de ventas y datos de clientes	5
2.2. Seguimiento de Venta de Productos	5
3. Subconsultas SELECT	6
3.1. Factura con productos	6
3.2. Seguimiento de ventas	7
4. Subconsultas IN	7
4.1. Rol de Administrador	7
4.2. Tienda con mayor stock	8
5. subconsultas EXISTS	8
5.1. stock de tiendas	8
5.2. Facturas asociadas a invoiceDetails	9
6. subconsultas ALL – ANY	9
6.1. Productos más caros en una categoría (ALL)	9
6.2. Facturas mayores que cualquier pago ANY	10

I. Introducción

Este informe presenta un análisis detallado de las diferentes subconsultas SQL implementadas en nuestro sistema de gestión comercial. Las subconsultas son herramientas fundamentales que nos permiten realizar análisis complejos y obtener información específica de nuestra base de datos.

Contexto y Relevancia

En el entorno actual de gestión comercial, la capacidad de extraer y analizar información específica de grandes volúmenes de datos es crucial para la toma de decisiones empresariales. Las subconsultas SQL se presentan como una solución robusta y flexible para este propósito, permitiendo realizar operaciones complejas que serían difíciles o imposibles de lograr con consultas simples.

Tipos de Subconsultas Analizadas

El informe abarca seis categorías principales de subconsultas:

1. **Subconsultas WHERE:** Utilizadas para filtrar resultados basados en condiciones complejas.
2. **Subconsultas FROM:** Permiten trabajar con conjuntos de datos derivados como si fueran tablas.
3. **Subconsultas SELECT:** Facilitan la obtención de valores calculados o relacionados para cada fila.
4. **Subconsultas IN:** Posibilitan la comparación con múltiples valores de manera eficiente.
5. **Subconsultas EXISTS:** Verifican la existencia de registros que cumplen ciertas condiciones.
6. **Subconsultas ALL/ANY:** Permiten comparaciones avanzadas contra conjuntos de valores.

Enlace de video:

<https://drive.google.com/file/d/1kyJ2FNKfMtMQ2Q4PAVmcetAAR97Az4JT/view?usp=sharing>

1. Subconsultas WHERE

1.1. Productos sin movimientos

```
-- 1. Encontrar productos que nunca se han vendido  
SELECT p.*  
FROM products p  
LEFT JOIN invoiceDetails id ON p.id = id.idProduct  
WHERE id.idProduct IS NULL;
```

RESULTADO:

id	name	code	price	priceDiscount	stock	idCategory	description	createdAt	updatedAt
----	------	------	-------	---------------	-------	------------	-------------	-----------	-----------

En el momento todos los productos se han vendido.

1.2. Clientes con gastos mayor al promedio

```
-- 2. Encontrar clientes que han gastado más que el promedio  
SELECT p.*  
FROM people p  
INNER JOIN (  
    SELECT idCustomer  
    FROM invoices  
    GROUP BY idCustomer  
    HAVING SUM(totalAmount) > (  
        SELECT AVG(totalAmount)  
        FROM invoices  
    )  
) AS avgSpenders ON p.id = avgSpenders.idCustomer;
```

Resultado:

	id	idUser	firstName	middleName	lastNameMaternal	lastNamePaternal	idDocumentType	document	phone	address	hiringDate	salary	createdAt	updatedAt
▶	1	1	John	Michael	Doe	Smith	1	123456789	1234567890	123 Main St	2023-01-15 00:00:00	5000	2024-10-21 21:11:25	2024-10-21 21:11:25
	4	4	Peter	James	Johnson	Brown	4	456789012	1234567893	101 Maple St	2023-04-30 00:00:00	3200	2024-10-21 21:11:25	2024-10-21 21:11:25
	5	5	Linda	Marie	Brown	White	5	567890123	1234567894	202 Birch St	2023-05-05 00:00:00	4000	2024-10-21 21:11:25	2024-10-21 21:11:25

2. Subconsultas FROM

2.1. Promedio de ventas y datos de clientes

-- 1. Obtener el promedio de ventas por cliente junto con sus datos personales

```
SELECT p.*, avgSales.promedioVentas
FROM people p
) INNER JOIN (
    SELECT idCustomer, AVG(totalAmount) as promedioVentas
    FROM invoices
    GROUP BY idCustomer
) avgSales ON p.id = avgSales.idCustomer;
```

RESULTADO:

	id	idUser	firstName	middleName	lastNameMaternal	lastNamePaternal	idDocumentType	document	phone	address	hiringDate	salary	createdAt	updatedAt	promedioVentas
▶	1	1	John	Michael	Doe	Smith	1	123456789	1234567890	123 Main St	2023-01-15 00:00:00	5000	2024-10-21 21:11:25	2024-10-21 21:11:25	1250.500000
	2	2	Jane	Elizabeth	Smith	Johnson	2	234567890	1234567891	456 Oak St	2023-02-20 00:00:00	3000	2024-10-21 21:11:25	2024-10-21 21:11:25	500.750000
	3	3	Maria	Carmen	Garcia	Lopez	3	345678901	1234567892	789 Pine St	2023-03-25 00:00:00	2500	2024-10-21 21:11:25	2024-10-21 21:11:25	799.990000
	4	4	Peter	James	Johnson	Brown	4	456789012	1234567893	101 Maple St	2023-04-30 00:00:00	3200	2024-10-21 21:11:25	2024-10-21 21:11:25	1200.000000
	5	5	Linda	Marie	Brown	White	5	567890123	1234567894	202 Birch St	2023-05-05 00:00:00	4000	2024-10-21 21:11:25	2024-10-21 21:11:25	999.490000

2.2. Seguimiento de Venta de Productos

-- 2. Obtener los productos más vendidos por categoría

```
SELECT pc.`name` as categoria, topProducts.*
FROM productsCategories pc
INNER JOIN (
    SELECT p.idCategory, p.`name`, COUNT(id.idProduct) as vecesVendido
    FROM products p
    INNER JOIN invoiceDetails id ON p.id = id.idProduct
    GROUP BY p.id, p.`name`, p.idCategory
) topProducts ON pc.id = topProducts.idCategory;
```

Resultado:

categoria	idCategory	name	vecesVendido
Lacteos	1	Leche	2
Granos	2	Arroz	1
Embutidos	3	Chorizo	1
Enlatados	4	Atún	2
Bebidas	5	Coca-Cola	1

3. Subconsultas SELECT

3.1. Factura con productos

-- 1. Mostrar cada factura con el total de productos diferentes que incluye

```
SELECT i.*, |
COUNT(idProduct) AS numProductosDiferentes
FROM invoices i
LEFT JOIN invoiceDetails id ON i.id = id.idInvoice
GROUP BY i.id;
```

RESULTADO:

id	invoiceDate	totalAmount	idCustomer	dataRespons	createdAt	updatedAt	numProductosDiferentes
1	2024-10-01 00:00:00	1250.50	1	Factura para John Doe, cliente regular	2024-10-21 21:11:25	2024-10-21 21:11:25	2
2	2024-10-02 00:00:00	500.75	2	Factura generada por compra con descuento	2024-10-21 21:11:25	2024-10-21 21:11:25	1
3	2024-10-03 00:00:00	799.99	3	Pago realizado en efectivo, cliente satisfecho	2024-10-21 21:11:25	2024-10-21 21:11:25	1
4	2024-10-04 00:00:00	1200.00	4	Factura urgente, se necesita entrega inmediata	2024-10-21 21:11:25	2024-10-21 21:11:25	1
5	2024-10-05 00:00:00	999.49	5	Factura procesada con PayPal	2024-10-21 21:11:25	2024-10-21 21:11:25	2

3.2. Seguimiento de ventas

-- 2. Mostrar cada producto con su última fecha de venta

```
SELECT p.*,  
(SELECT MAX(invoiceDate)  
FROM invoices i  
INNER JOIN invoiceDetails id ON i.id = id.idInvoice  
WHERE id.idProduct = p.id) AS ultima_venta  
FROM products p;
```

RESULTADO:

id	name	code	price	priceDiscount	stock	idCategory	description	createdAt	updatedAt	ultimaventa
1	Leche	ABC123	999.99	899.99	10	1	Bolsa de leche marca Colanta	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-05 00:00:00
2	Arroz	DEF456	19.99	14.99	100	2	Arroz marca Diana	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-01 00:00:00
3	Chorizo	GHI789	1.50	1.00	500	3	Chorizos Zenú	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-02 00:00:00
4	Atún	JKL012	12.99	9.99	50	4	Atún marca Isabell	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-05 00:00:00
5	Coca-Cola	MNO345	199.99	149.99	20	5	Coca-Cola	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-04 00:00:00

4. Subconsultas IN

4.1. Rol de Administrador

-- 1. Encontrar todos los usuarios que tienen rol de administrador

```
SELECT *  
FROM users  
WHERE id IN (  
SELECT idUser  
FROM usersRoles  
WHERE idRole IN (  
SELECT id  
FROM roles  
WHERE name = 'Administrador'  
)  
);
```

RESULTADO:

id	email	name	password	createdAt	updatedAt
1	john.doe@example.com	John Doe	jhon_password_123	2024-10-21 21:11:25	2024-10-21 21:11:25
NULL	NULL	NULL	NULL	NULL	NULL

4.2. Tienda con mayor stock

```
-- 2. Encontrar productos que están en tiendas con stock mayor a 100
SELECT *
FROM products
WHERE id IN (
    SELECT idProduct
    FROM storeProducts
    WHERE stock > 100
);
```

RESULTADO:

	id	name	code	price	priceDiscount	stock	idCategory	description	createdAt	updatedAt
►	3	Chorizo	GHI789	1.50	1.00	500	3	Chorizos Zenú	2024-10-21 21:11:25	2024-10-21 21:11:25
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. subconsultas EXISTS

5.1. stock de tiendas

```
-- Encontrar tiendas que tienen al menos un producto con stock
SELECT *
FROM store s
WHERE EXISTS (
    SELECT 1
    FROM storeProducts sp
    WHERE sp.idStore = s.id
    AND sp.stock > 0
);
```

RESULTADOS:

Informe de Subconsultas SQL

Presentado por:

Michael Alejandro Papamija Pantoja

Docente:

Brayan Arcos

Materia:

Desarrollo de Base de Datos

Instituto Tecnológico del Putumayo

Mocoa – Putumayo

2024

Tabla de contenido

I. Introducción	3
Contexto y Relevancia	3
Tipos de Subconsultas Analizadas	3
1. Subconsultas WHERE	4
1.1. Productos sin movimientos	4
1.2. Clientes con gastos mayor al promedio	4
2. Subconsultas FROM	5
2.1. Promedio de ventas y datos de clientes	5
2.2. Seguimiento de Venta de Productos	5
3. Subconsultas SELECT	6
3.1. Factura con productos	6
3.2. Seguimiento de ventas	7
4. Subconsultas IN	7
4.1. Rol de Administrador	7
4.2. Tienda con mayor stock	8
5. subconsultas EXISTS	8
5.1. stock de tiendas	8
5.2. Facturas asociadas a invoiceDetails	9
6. subconsultas ALL – ANY	9
6.1. Productos más caros en una categoría (ALL)	9
6.2. Facturas mayores que cualquier pago ANY	10

I. Introducción

Este informe presenta un análisis detallado de las diferentes subconsultas SQL implementadas en nuestro sistema de gestión comercial. Las subconsultas son herramientas fundamentales que nos permiten realizar análisis complejos y obtener información específica de nuestra base de datos.

Contexto y Relevancia

En el entorno actual de gestión comercial, la capacidad de extraer y analizar información específica de grandes volúmenes de datos es crucial para la toma de decisiones empresariales. Las subconsultas SQL se presentan como una solución robusta y flexible para este propósito, permitiendo realizar operaciones complejas que serían difíciles o imposibles de lograr con consultas simples.

Tipos de Subconsultas Analizadas

El informe abarca seis categorías principales de subconsultas:

1. **Subconsultas WHERE:** Utilizadas para filtrar resultados basados en condiciones complejas.
2. **Subconsultas FROM:** Permiten trabajar con conjuntos de datos derivados como si fueran tablas.
3. **Subconsultas SELECT:** Facilitan la obtención de valores calculados o relacionados para cada fila.
4. **Subconsultas IN:** Posibilitan la comparación con múltiples valores de manera eficiente.
5. **Subconsultas EXISTS:** Verifican la existencia de registros que cumplen ciertas condiciones.
6. **Subconsultas ALL/ANY:** Permiten comparaciones avanzadas contra conjuntos de valores.

Enlace de video:

<https://drive.google.com/file/d/1kyJ2FNKfMtMQ2Q4PAVmcetAAR97Az4JT/view?usp=sharing>

1. Subconsultas WHERE

1.1. Productos sin movimientos

```
-- 1. Encontrar productos que nunca se han vendido  
SELECT p.*  
FROM products p  
LEFT JOIN invoiceDetails id ON p.id = id.idProduct  
WHERE id.idProduct IS NULL;
```

RESULTADO:

id	name	code	price	priceDiscount	stock	idCategory	description	createdAt	updatedAt
----	------	------	-------	---------------	-------	------------	-------------	-----------	-----------

En el momento todos los productos se han vendido.

1.2. Clientes con gastos mayor al promedio

```
-- 2. Encontrar clientes que han gastado más que el promedio  
SELECT p.*  
FROM people p  
INNER JOIN (  
    SELECT idCustomer  
    FROM invoices  
    GROUP BY idCustomer  
    HAVING SUM(totalAmount) > (  
        SELECT AVG(totalAmount)  
        FROM invoices  
    )  
) AS avgSpenders ON p.id = avgSpenders.idCustomer;
```

Resultado:

	id	idUser	firstName	middleName	lastNameMaternal	lastNamePaternal	idDocumentType	document	phone	address	hiringDate	salary	createdAt	updatedAt
▶	1	1	John	Michael	Doe	Smith	1	123456789	1234567890	123 Main St	2023-01-15 00:00:00	5000	2024-10-21 21:11:25	2024-10-21 21:11:25
	4	4	Peter	James	Johnson	Brown	4	456789012	1234567893	101 Maple St	2023-04-30 00:00:00	3200	2024-10-21 21:11:25	2024-10-21 21:11:25
	5	5	Linda	Marie	Brown	White	5	567890123	1234567894	202 Birch St	2023-05-05 00:00:00	4000	2024-10-21 21:11:25	2024-10-21 21:11:25

2. Subconsultas FROM

2.1. Promedio de ventas y datos de clientes

-- 1. Obtener el promedio de ventas por cliente junto con sus datos personales

```
SELECT p.*, avgSales.promedioVentas
FROM people p
INNER JOIN (
    SELECT idCustomer, AVG(totalAmount) as promedioVentas
    FROM invoices
    GROUP BY idCustomer
) avgSales ON p.id = avgSales.idCustomer;
```

RESULTADO:

	id	idUser	firstName	middleName	lastNameMaternal	lastNamePaternal	idDocumentType	document	phone	address	hiringDate	salary	createdAt	updatedAt	promedioVentas
▶	1	1	John	Michael	Doe	Smith	1	123456789	1234567890	123 Main St	2023-01-15 00:00:00	5000	2024-10-21 21:11:25	2024-10-21 21:11:25	1250.500000
	2	2	Jane	Elizabeth	Smith	Johnson	2	234567890	1234567891	456 Oak St	2023-02-20 00:00:00	3000	2024-10-21 21:11:25	2024-10-21 21:11:25	500.750000
	3	3	Maria	Carmen	Garcia	Lopez	3	345678901	1234567892	789 Pine St	2023-03-25 00:00:00	2500	2024-10-21 21:11:25	2024-10-21 21:11:25	799.990000
	4	4	Peter	James	Johnson	Brown	4	456789012	1234567893	101 Maple St	2023-04-30 00:00:00	3200	2024-10-21 21:11:25	2024-10-21 21:11:25	1200.000000
	5	5	Linda	Marie	Brown	White	5	567890123	1234567894	202 Birch St	2023-05-05 00:00:00	4000	2024-10-21 21:11:25	2024-10-21 21:11:25	999.490000

2.2. Seguimiento de Venta de Productos

-- 2. Obtener los productos más vendidos por categoría

```
SELECT pc.`name` as categoria, topProducts.*
FROM productsCategories pc
INNER JOIN (
    SELECT p.idCategory, p.`name`, COUNT(id.idProduct) as vecesVendido
    FROM products p
    INNER JOIN invoiceDetails id ON p.id = id.idProduct
    GROUP BY p.id, p.`name`, p.idCategory
) topProducts ON pc.id = topProducts.idCategory;
```

Resultado:

categoria	idCategory	name	vecesVendido
Lacteos	1	Leche	2
Granos	2	Arroz	1
Embutidos	3	Chorizo	1
Enlatados	4	Atún	2
Bebidas	5	Coca-Cola	1

3. Subconsultas SELECT

3.1. Factura con productos

-- 1. Mostrar cada factura con el total de productos diferentes que incluye

```
SELECT i.*, |
COUNT(idProduct) AS numProductosDiferentes
FROM invoices i
LEFT JOIN invoiceDetails id ON i.id = id.idInvoice
GROUP BY i.id;
```

RESULTADO:

id	invoiceDate	totalAmount	idCustomer	dataRespons	createdAt	updatedAt	numProductosDiferentes
1	2024-10-01 00:00:00	1250.50	1	Factura para John Doe, cliente regular	2024-10-21 21:11:25	2024-10-21 21:11:25	2
2	2024-10-02 00:00:00	500.75	2	Factura generada por compra con descuento	2024-10-21 21:11:25	2024-10-21 21:11:25	1
3	2024-10-03 00:00:00	799.99	3	Pago realizado en efectivo, cliente satisfecho	2024-10-21 21:11:25	2024-10-21 21:11:25	1
4	2024-10-04 00:00:00	1200.00	4	Factura urgente, se necesita entrega inmediata	2024-10-21 21:11:25	2024-10-21 21:11:25	1
5	2024-10-05 00:00:00	999.49	5	Factura procesada con PayPal	2024-10-21 21:11:25	2024-10-21 21:11:25	2

3.2. Seguimiento de ventas

-- 2. Mostrar cada producto con su última fecha de venta

```
SELECT p.*,  
(SELECT MAX(invoiceDate)  
FROM invoices i  
INNER JOIN invoiceDetails id ON i.id = id.idInvoice  
WHERE id.idProduct = p.id) AS ultima_venta  
FROM products p;
```

RESULTADO:

id	name	code	price	priceDiscount	stock	idCategory	description	createdAt	updatedAt	ultimaventa
1	Leche	ABC123	999.99	899.99	10	1	Bolsa de leche marca Colanta	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-05 00:00:00
2	Arroz	DEF456	19.99	14.99	100	2	Arroz marca Diana	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-01 00:00:00
3	Chorizo	GHI789	1.50	1.00	500	3	Chorizos Zenú	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-02 00:00:00
4	Atún	JKL012	12.99	9.99	50	4	Atún marca Isabell	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-05 00:00:00
5	Coca-Cola	MNO345	199.99	149.99	20	5	Coca-Cola	2024-10-21 21:11:25	2024-10-21 21:11:25	2024-10-04 00:00:00

4. Subconsultas IN

4.1. Rol de Administrador

-- 1. Encontrar todos los usuarios que tienen rol de administrador

```
SELECT *  
FROM users  
WHERE id IN (  
SELECT idUser  
FROM usersRoles  
WHERE idRole IN (  
SELECT id  
FROM roles  
WHERE name = 'Administrador'  
)  
);
```

RESULTADO:

id	email	name	password	createdAt	updatedAt
1	john.doe@example.com	John Doe	jhon_password_123	2024-10-21 21:11:25	2024-10-21 21:11:25
NULL	NULL	NULL	NULL	NULL	NULL

4.2. Tienda con mayor stock

```
-- 2. Encontrar productos que están en tiendas con stock mayor a 100
SELECT *
FROM products
WHERE id IN (
    SELECT idProduct
    FROM storeProducts
    WHERE stock > 100
);
```

RESULTADO:

	id	name	code	price	priceDiscount	stock	idCategory	description	createdAt	updatedAt
▶	3	Chorizo	GHI789	1.50	1.00	500	3	Chorizos Zenú	2024-10-21 21:11:25	2024-10-21 21:11:25
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. subconsultas EXISTS

5.1. stock de tiendas

```
-- Encontrar tiendas que tienen al menos un producto con stock
SELECT *
FROM store s
WHERE EXISTS (
    SELECT 1
    FROM storeProducts sp
    WHERE sp.idStore = s.id
    AND sp.stock > 0
);
```

RESULTADOS:

