Juan José Gallego – Juan José Hernández – Nicolas Rojas

Diseñar consultas SQL que respondan preguntas de negocio:

1. Total de ventas por categoría de producto:

```
SELECT p. product AS product,

    SUM(i.total_price) AS total_sales

FROM invoice i

JOIN Product p ON i.product_id p. product_id

GROUP BY p.product

ORDER BY total_sales DESC;
```

Esta consulta une la tabla invoice y product por medio del id, luego selecciona las columnas categoría y total_ventas las cuales muestran cuánto generó cada categoría. También agrupa las ventas totales por cada categoría y se ordena para mostrar qué categoría generó más.

2. cliente con mayor volumen (dinero) de compras

```
SELECT customer_id,
SUM(total_price) AS total_purchases
FROM invoice GROUP BY customer_id
ORDER BY total_purchases

DESC LIMIT 1;
```

a cada cliente le suma el total de compras en todas sus compras realizadas y retorna el cliente con mayor volumen de dinero gastado en las tiendas

3. Métodos de pago más utilizados:

```
SELECT p.payment_method AS payment_method,

    COUNT(i.payment_id) AS times_used

FROM invoice i
```

JOIN payment_method p ON i.payment_id = p.payment_id

GROUP BY p. payment_method

ORDER BY times_used DESC;

Esta consulta agrupa las tablas payment_method y invoice, selecciona las columnas method_name y cuenta el payment_id para generar un atributo llamado times_used,

luego agrupa los resultados por el method_name y ordenar descendentemente para identificar cuál método fue el más utilizado.

4.

```
SELECT
   d.year,
   d.month,
   SUM(i.total_price) AS total_sales,
   COUNT(i.invoice_no) AS sales_quantity
FROM invoice i
JOIN Date d ON i.date_id = d.date_id
GROUP BY d.year, d.month
ORDER BY d.year, d.month;
```

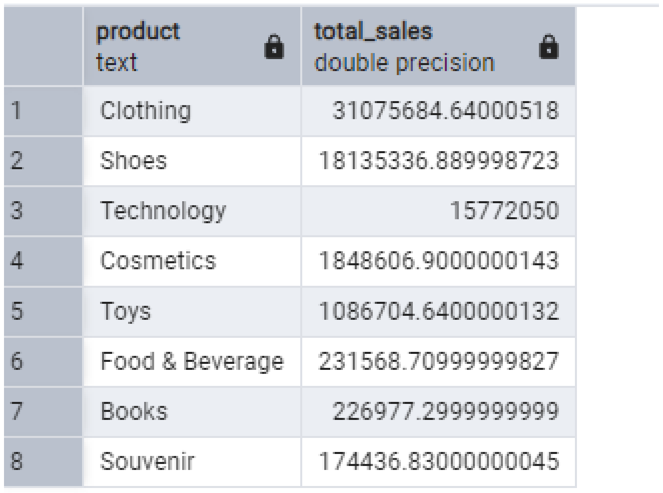
Genera una tabla donde cada mes tiene el número total de ventas y el acumulado de dinero de estas ventas

Optimizar las consultas utilizando índices y agregaciones.

1.

```
-- Crear índices para optimizar la consulta
CREATE INDEX idx_invoice_product ON invoice(product_id);
CREATE INDEX idx_invoice_total_price ON invoice(total_price);
CREATE INDEX idx_product_product_id ON Product(product_id);

-- Consulta optimizada
SELECT p.product AS product,
     SUM(i.total_price) AS total_sales
```

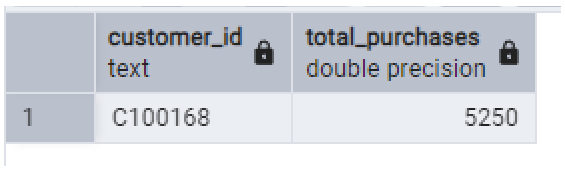
```
FROM invoice i
JOIN product p ON i.product_id = p.product_id
GROUP BY p.product_id, p.product
ORDER BY total_sales DESC;
```



	product text	total_sales double precision
1	Clothing	31075684.64000518
2	Shoes	18135336.889998723
3	Technology	15772050
4	Cosmetics	1848606.9000000143
5	Toys	1086704.6400000132
6	Food & Beverage	231568.70999999827
7	Books	226977.2999999999
8	Souvenir	174436.83000000045

2.

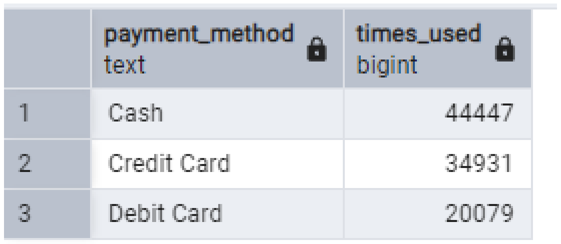
```
-- Crear índices para mejorar el rendimiento

CREATE INDEX idx_invoice_customer ON invoice(customer_id);

CREATE INDEX idx_invoice_total_price ON invoice(total_price);


-- Consulta optimizada para obtener el cliente con más compras

SELECT customer_id,

    SUM(total_price) AS total_purchases

FROM invoice

GROUP BY customer_id

ORDER BY total_purchases DESC

LIMIT 1;
```



	customer_id text	total_purchases double precision
1	C100168	5250

3.

```
-- Crear índices para optimizar la consulta

CREATE INDEX idx_invoice_payment_id ON invoice(payment_id);

CREATE INDEX idx_payment_method_id ON payment_method(payment_id);


-- Consulta optimizada para contar los métodos de pago más usados

SELECT p.payment_method AS payment_method,

    COUNT(i.payment_id) AS times_used

FROM invoice i

JOIN payment_method p ON i.payment_id = p.payment_id

GROUP BY p.payment_id, p.payment_method

ORDER BY times_used DESC;
```

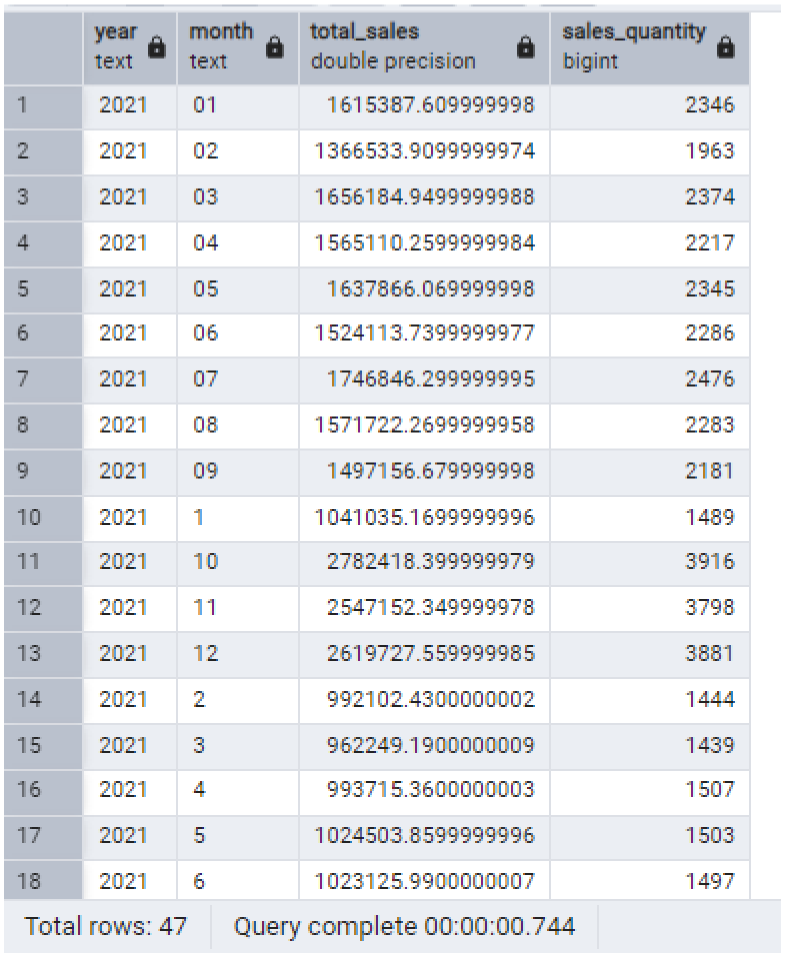


	payment_method text	times_used bigint
1	Cash	44447
2	Credit Card	34931
3	Debit Card	20079

4.

```
-- Crear índices para mejorar la consulta
CREATE INDEX idx_invoice_date_id ON invoice(date_id);
CREATE INDEX idx_invoice_total_price ON invoice(total_price);
CREATE INDEX idx_invoice_invoice_no ON invoice(invoice_no);
CREATE INDEX idx_date_id ON Date(date_id);
CREATE INDEX idx_date_year_month ON Date(year, month);

-- Consulta optimizada
SELECT
   d.year,
   d.month,
   SUM(i.total_price) AS total_sales,
   COUNT(i.invoice_no) AS sales_quantity
FROM invoice i
JOIN Date d ON i.date_id = d.date_id
GROUP BY d.year, d.month
```

```
ORDER BY d.year, d.month;
```



	year text	month text	total_sales double precision	sales_quantity bigint
1	2021	01	1615387.609999998	2346
2	2021	02	1366533.9099999974	1963
3	2021	03	1656184.9499999988	2374
4	2021	04	1565110.2599999984	2217
5	2021	05	1637866.069999998	2345
6	2021	06	1524113.7399999977	2286
7	2021	07	1746846.299999995	2476
8	2021	08	1571722.2699999958	2283
9	2021	09	1497156.679999998	2181
10	2021	1	1041035.1699999996	1489
11	2021	10	2782418.399999979	3916
12	2021	11	2547152.349999978	3798
13	2021	12	2619727.559999985	3881
14	2021	2	992102.4300000002	1444
15	2021	3	962249.1900000009	1439
16	2021	4	993715.3600000003	1507
17	2021	5	1024503.8599999996	1503
18	2021	6	1023125.9900000007	1497

Total rows: 47 Query complete 00:00:00.744