Segmentez des clients d'un site e-commerce

```
--Afficher chaque table
SELECT * FROM customers;
SELECT * FROM geoloc;
SELECT * FROM order_items;
SELECT * FROM order_pymts;
SELECT * FROM order_reviews;
SELECT * FROM orders;
SELECT * FROM products;
SELECT * FROM sellers;
SELECT * FROM translation;
--comment inspecter les colonnes d'une table
PRAGMA table_info(products);
```

/*En excluant les commandes annulées, quelles sont les commandes récentes de moins de 3 mois que les clients ont reçues avec au moins 3 jours de retard ?*/

```
SELECT
  "index",
  order_id,
  customer_id,
  order_status,
  order_purchase_timestamp,
  order_approved_at,
  order_delivered_carrier_date,
  order_delivered_customer_date,
  order_estimated_delivery_date,
  julianday(order_delivered_customer_date) -
julianday(order_estimated_delivery_date) AS delivery_delay
FROM
  orders
WHERE
  order_status != 'canceled'
  AND order_purchase_timestamp >= date('2018-10-17', '-3 months')
  AND julianday(order_delivered_customer_date) -
julianday(order_estimated_delivery_date) >= 3
ORDER BY
  order_purchase_timestamp DESC
LIMIT 10;
```

/*Qui sont les vendeurs ayant généré un chiffre d'affaires de plus de 100 000 Real sur des commandes livrées via Olist ?*/

```
WITH table1 as (
SELECT DISTINCT oi.seller_id, oi.order_id, oi.price
FROM order_items as oi
JOIN orders as o
ON o.order_id = oi.order_id
WHERE o.order_status = "delivered"
),
table2 as (
SELECT seller_id, sum(price) as produit_total
FROM table1
GROUP BY seller_id
)
SELECT * FROM table2
WHERE produit_total > 100000
ORDER BY
produit_total;
```

/* Qui sont les nouveaux vendeurs (moins de 3 mois d'ancienneté) qui sont déjà très engagés avec la plateforme (ayant déjà vendu plus de 30 produits) ?*/

```
WITH table1 as (
SELECT DISTINCT oi.seller_id, oi.order_id, o.order_purchase_timestamp
FROM order_items as oi
JOIN orders as o
ON o.order_id = oi.order_id
WHERE o.order_status = "delivered"
),
table2 as (
SELECT max (o.order_purchase_timestamp) AS date_max
FROM orders AS o),
table3 AS (
SELECT seller_id, count (order_id) AS nombre_total_id
FROM table1
GROUP BY seller_id
HAVING min (order_purchase_timestamp) > date ((SELECT date_max FROM
table2), "-3 months")
)
SELECT *FROM table3
```

```
WHERE nombre_total_id > 30;
```

/quels sont les 5 codes postaux, enregistrant plus de 30 reviews, avec le pire review score moyen sur les 12 derniers mois ?*/

```
WITH table1 as (
SELECT max (o.order_purchase_timestamp) AS date_max
FROM orders AS o),
table2 AS (
SELECT DISTINCT o.order_id, table_client.customer_zip_code_prefix,
o.customer_id, o.order_purchase_timestamp
FROM orders AS o JOIN customers AS table_client
ON o.customer_id = table_client.customer_id
),
table3 AS (
SELECT customer_zip_code_prefix, AVG(review_score) AS moyenne_avis,
COUNT(review_score) AS nobre_avis
FROM order_reviews AS score_revieu JOIN table2
ON  table2.order_id = score_revieu.order_id
WHERE order_purchase_timestamp  >  date((SELECT date_max FROM table1),
"-12 months")
GROUP BY customer_zip_code_prefix
)
SELECT * FROM table3
WHERE nobre_avis > 30
order by moyenne_avis
LIMIT 5;
```