```sql
/*
● 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 order_id, customer_id, order_purchase_timestamp, order_estimated_delivery_date, order_delivered_customer_date
FROM orders
WHERE order_status != 'canceled'
AND order_purchase_timestamp >= date('2018-11-12 00:00:00', '-3 month')
AND julianday(order_delivered_customer_date) - julianday(order_estimated_delivery_date) >= 3

/*
● Qui sont les vendeurs ayant généré un chiffre d'affaires de plus de 100
000 Real sur des commandes livrées via Olist ?
*/
SELECT seller_id, ROUND(SUM(price), 2) AS total_revenue
FROM order_items
GROUP BY seller_id
HAVING ROUND(SUM(price), 2) > 100000;

/*
● 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 SellerFirstSale AS (
    SELECT oi.seller_id, MIN(o.order_purchase_timestamp) as FirstSaleDate
    FROM order_items oi
    JOIN orders o ON oi.order_id = o.order_id
    GROUP BY oi.seller_id
),
SellerSalesCount AS (
    SELECT oi.seller_id, COUNT(oi.order_item_id) as TotalSales
    FROM order_items oi
    JOIN orders o ON oi.order_id = o.order_id
    WHERE o.order_purchase_timestamp <= '2018-08-29 15:00:37'
    AND o.order_purchase_timestamp >= DATE('2018-08-29 15:00:37', '-3 months')
    GROUP BY oi.seller_id
)
SELECT ssc.seller_id
FROM SellerSalesCount ssc
JOIN SellerFirstSale sfs ON ssc.seller_id = sfs.seller_id
WHERE sfs.FirstSaleDate >= DATE('2018-08-29 15:00:37', '-3 months')
AND ssc.TotalSales > 30;

/*
● Quels sont les 5 codes postaux, enregistrant plus de 30 commandes,
avec le pire review score moyen sur les 12 derniers mois ?
*/
WITH FilteredOrders AS (
    SELECT o.order_id, c.customer_zip_code_prefix
    FROM orders o
    JOIN customers c ON o.customer_id = c.customer_id
    WHERE o.order_purchase_timestamp BETWEEN DATE('2018-08-29 15:00:37', '-12 months') AND '2018-08-29 15:00:37'
),
ReviewScores AS (
    SELECT fo.customer_zip_code_prefix, ROUND(AVG(orev.review_score), 2) as AvgReviewScore, COUNT(fo.order_id) as OrderCount
    FROM FilteredOrders fo
    JOIN order_reviews orev ON fo.order_id = orev.order_id
    GROUP BY fo.customer_zip_code_prefix
    HAVING OrderCount > 30
)
SELECT customer_zip_code_prefix, AvgReviewScore
FROM ReviewScores
ORDER BY AvgReviewScore ASC
LIMIT 5;
```