

requête 1 : 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 ?

La requête retourne 322 lignes

olist(1).db

req4.sql

req3.sql

req2.sql

req1.sql



```
1 SELECT *,
2     JULIANDAY(order_delivered_customer_date) - JULIANDAY(order_estimated_delivery_date) AS delta
3 FROM orders
4 WHERE order_purchase_timestamp >= (SELECT DATE(MAX(order_purchase_timestamp), '-3 months') FROM orders)
5    AND JULIANDAY(order_delivered_customer_date) - JULIANDAY(order_estimated_delivery_date) >= 3
6    AND order_status <> 'canceled';
```

index	order_id	customer_id	order_status	order_purcha...	order_approv...	order_deliver...	order_deliver...	order_estimat...	delta
561	cfa4fa27b41797...	7093250e1741e...	delivered	2018-08-16 09:4...	2018-08-17 03:1...	2018-08-17 18:5...	2018-08-29 01:4...	2018-08-22 00:0...	7.070613426156342
711	234c056c50619f...	44e460a655f715...	delivered	2018-08-14 14:4...	2018-08-15 03:1...	2018-08-31 15:2...	2018-09-01 18:1...	2018-08-23 00:0...	9.760208333376795
728	8ad3f1d0f96992...	948b29e24216a...	delivered	2018-07-17 21:2...	2018-07-17 21:3...	2018-07-18 13:0...	2018-08-14 04:0...	2018-08-03 00:0...	11.169907407369465
1844	7f579e203c931f...	d665be250d1c6...	delivered	2018-08-02 18:0...	2018-08-02 18:4...	2018-08-10 14:1...	2018-08-13 20:1...	2018-08-09 00:0...	4.841516203712672
1967	cb6e441ff2ef574...	4fb843d304c571...	delivered	2018-08-08 19:2...	2018-08-08 19:4...	2018-08-10 13:3...	2018-08-18 01:1...	2018-08-15 00:0...	3.0499768517911434
2349	03720fdc92032e...	116458665bac0f...	delivered	2018-08-05 21:3...	2018-08-05 21:4...	2018-08-15 17:4...	2018-08-21 00:1...	2018-08-17 00:0...	4.00824074074626
2786	7a268da1c6173...	293407b5065d8f...	delivered	2018-07-23 19:1...	2018-07-24 10:3...	2018-07-24 13:5...	2018-08-14 17:2...	2018-08-10 00:0...	4.7232870371080935
3177	4c8c092a89740...	a82a9ed42b002...	delivered	2018-08-07 08:4...	2018-08-07 08:5...	2018-08-13 12:4...	2018-08-14 17:5...	2018-08-10 00:0...	4.743981481529772
3245	1cf0fc2c07de89...	97a0e2c05d588...	delivered	2018-08-07 21:5...	2018-08-07 22:0...	2018-08-17 14:0...	2018-08-18 18:2...	2018-08-15 00:0...	3.764710647985339
3328	34871174fe1f4c...	0d72c06cbd17c...	delivered	2018-08-19 22:4...	2018-08-21 04:3...	2018-08-23 13:4...	2018-08-30 16:0...	2018-08-27 00:0...	3.6687731482088566

requête 2 : Qui sont les vendeurs ayant généré un chiffre d'affaires de plus de 100 000 Real sur des commandes livrées via Olist ?

La requête retourne 15 lignes

olist(1).db

req4.sql

req3.sql

req2.sql

req1.sql



```
1 SELECT seller_id,
2        SUM(price) AS total_amount_sold,
3        COUNT(order_id) AS total_items_sold
4 FROM (
5     SELECT DISTINCT i.seller_id,
6                    i.order_id,
7                    i.price
8     FROM order_items AS i
9     INNER JOIN orders AS o ON o.order_id = i.order_id
10    WHERE o.order_status = 'delivered'
11 ) AS orders_joined
12 GROUP BY seller_id
13 HAVING SUM(price) > 100000
14 ORDER BY total_amount_sold DESC;
```

seller_id	total_amount_sold	total_items_sold
4869f7a5dfa277a7dca6462dcf3b52b2	224921.54	1135
53243585a1d6dc2643021fd1853d8905	203364	348
fa1c13f2614d7b5c4749cbc52fecda94	190917.14	579
4a3ca9315b744ce9f8e9374361493884	183828.07	1810
7e93a43ef30c4f03f38b393420bc753a	165822.49	321
da8622b14eb17ae2831f4ac5b9dab84a	141875.38	1371
7c67e1448b00f6e969d365cea6b010ab	140998.62	995
7a67c85e85bb2ce8582c35f2203ad736	139098.77	1147
46dc3b2cc0980fb8ec44634e21d2718e	120671.54000000001	504
6560211a19b47992c3666cc44a7e94c0	116660.83	1905



requête 3 : 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) ?

La requête retourne 3 lignes

olist(1).db req4.sql req3.sql req2.sql req1.sql olist(1).db.2

```
1 WITH latest_order AS (  
2     SELECT MAX(order_purchase_timestamp)  
3     FROM orders),  
4 orders_joined AS (  
5     SELECT DISTINCT i.seller_id, i.order_id, i.price, o.order_purchase_timestamp  
6     FROM order_items AS i  
7     INNER JOIN orders AS o  
8     ON o.order_id = i.order_id  
9     WHERE o.order_status = 'delivered'  
10    ORDER BY seller_id, order_purchase_timestamp),  
11 aggregation AS (  
12    SELECT seller_id, sum(price) AS total_amount_sold, COUNT(order_id) AS total_items_sold  
13    FROM orders_joined  
14    GROUP BY seller_id  
15    HAVING min(order_purchase_timestamp) > date((SELECT * FROM latest_order), '-3 months'))  
16 SELECT* FROM aggregation  
17 WHERE total_items_sold > 30
```

! seller_id	total_amount_sold	total_items_sold
240b9776d844d37535668549a396af32	13332.09	35
81f89e42267213cb94da7ddc301651da	3522	46
d13e50eaa47b4cbe9eb81465865d8cfc	6987.15	67

requête 4 : Quels sont les 5 codes postaux, enregistrant plus de 30 reviews, avec le pire review score moyen sur les 12 derniers mois ?

La requête retourne 5 lignes

olist(1).db

req4.sql

req3.sql

req2.sql

req1.sql



```
1 SELECT
2     c.customer_zip_code_prefix,
3     AVG("or".review_score) AS avg_review_score,
4     COUNT("or".review_score) AS nb_reviews
5 FROM
6     "order_reviews" AS "or"
7 INNER JOIN
8     "orders" AS o ON "or".order_id = o.order_id
9 INNER JOIN
10    "customers" AS c ON o.customer_id = c.customer_id
11 WHERE
12     o.order_purchase_timestamp >= (SELECT MAX(order_purchase_timestamp, '-12 months') FROM orders)
13 GROUP BY
14     c.customer_zip_code_prefix
15 HAVING
16     nb_reviews > 30
17 ORDER BY
18     avg_review_score
19 LIMIT 5
```

customer_zip_code_prefix	avg_review_score	nb_reviews
22753	2.8085106382978724	47
22770	3.1842105263157894	38
13056	3.21875	32
22793	3.2717391304347827	92
21321	3.282051282051282	39

