

☰ -- Qual a quantidade de cidades únicas de todos os vendedores da base?
SELECT
COUNT(DISTINCT s.seller_city) AS unique_seller_city
FROM sellers s

	123 unique_seller_city ▼
1	611

☰ -- Qual o valor médio do preço de todos os pedidos?
SELECT
ROUND(AVG(price),2) AS avg_price
FROM order_items oi

	123 avg_price ▼
1	120.65

☰ -- Qual o maior valor de preço entre todos os pedidos?
SELECT
MAX(oi.price) AS max_price
FROM order_items oi

	123 max_price ▼
1	6,735

☰ -- Qual o menor valor de preço entre todos os pedidos?
SELECT
MIN(oi.price) AS min_price
FROM order_items oi

	123 min_price ▼
1	0.85

☰ -- Qual a quantidade de produtos distintos vendidos?
SELECT
COUNT(DISTINCT oi.product_id) AS unique_products
FROM order_items oi

	123 unique_products ▼
1	32,951

```
-- Quais os tipos de pagamentos existentes?
SELECT
    DISTINCT op.payment_type AS payment_types
FROM order_payments op
```

	ABC payment_types	
1	credit_card	
2	boleto	
3	voucher	
4	debit_card	
5	not_defined	

```
-- Qual o maior número de parcelas realizado?
SELECT
    MAX(op.payment_installments) AS max_payment_installments
FROM order_payments op
```

	123 max_payment_installments	
1	24	

```
-- Qual o menor número de parcelas realizado?
SELECT
    MIN(op.payment_installments) AS min_payment_installments
FROM order_payments op
```

	123 min_payment_installments	
1	0	

```
-- Qual a média do valor pago no cartão de crédito?
SELECT
    ROUND(AVG(op.payment_value),2) AS avg_credit_payment
FROM order_payments op
WHERE op.payment_type = 'credit_card'
```

	123 avg_credit_payment	
1	163.32	

● -- Quais os tipos de status para um pedido?
SELECT
 DISTINCT o.order_status **AS** unique_status
FROM orders o

	ABC unique_status ▼	
1	delivered	
2	invoiced	
3	shipped	
4	processing	
5	unavailable	
6	canceled	
7	created	
8	approved	

● -- Quantos clientes distintos fizeram um pedido?
SELECT
 COUNT(**DISTINCT** o.customer_id) **AS** unique_customer_qty
FROM orders o

	🔒 123 unique_customer_qty ▼	
1	99,441	

● -- Qual o número de clientes únicos de todos os estados?
SELECT
 c.customer_state,
 COUNT(**DISTINCT** c.customer_id) **AS** clientes_unicos
FROM customer c
GROUP BY c.customer_state
ORDER BY clientes_unicos **DESC**

	customer_state	clientes_unicos
1	SP	41,746
2	RJ	12,852
3	MG	11,635
4	RS	5,466
5	PR	5,045
6	SC	3,637
7	BA	3,380
8	DF	2,140
9	ES	2,033
10	GO	2,020
11	PE	1,652
12	CE	1,336
13	PA	975
14	MT	907
15	MA	747
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

```
-- Qual o número de cidades únicas de todos os estados?
SELECT
    g.geolocation_state,
    COUNT(DISTINCT g.geolocation_city) AS numero_cidades
FROM geolocation g
GROUP BY g.geolocation_state
ORDER BY numero_cidades DESC
```

	geolocation_state	numero_cidades
1	MG	1,426
2	SP	1,048
3	RS	691
4	BA	652
5	PR	651
6	SC	420
7	GO	384
8	MA	299
9	PI	278
10	PE	267
11	CE	260
12	PB	254
13	RJ	245
14	PA	219
15	RN	214
16	MT	213
17	TO	173
18	ES	160
19	MS	133
20	AL	130
21	SE	96
22	RO	83
23	AM	74
24	AC	34
25	DF	28
26	AP	17
27	RR	14

```
-- Qual o número de clientes únicos por estado e por cidade?
SELECT
    c.customer_state,
    c.customer_city,
    COUNT(DISTINCT c.customer_id) AS clientes_unicos
FROM customer c
GROUP BY c.customer_state, c.customer_city
```

	ABC customer_state	ABC customer_city	123 clientes_unicos	
1	AC	brasileia	1	
2	AC	cruzeiro do sul	3	
3	AC	epitaciolandia	1	
4	AC	manoel urbano	1	
5	AC	porto acre	1	
6	AC	rio branco	70	
7	AC	senador guiomard	2	
8	AC	xapuri	2	
9	AL	agua branca	1	
10	AL	anadia	2	
11	AL	arapiraca	29	
12	AL	atalaia	1	
13	AL	barra de santo antonio	2	
14	AL	barra de sao miguel	2	
15	AL	batalha	3	
16	AL	belem	3	
17	AL	boca da mata	2	
18	AL	cacimbinhas	1	
19	AL	cajueiro	1	
20	AL	campo alegre	2	
21	AL	canapi	1	
22	AL	coite do noia	1	
23	AL	colonia leopoldina	2	
24	AL	coruripe	3	
25	AL	delmiro gouveia	4	
26	AL	dois riachos	1	

```

• -- Qual a quantidade de pedidos, a média do valor do pagamento e o número máximo de parcelas por tipo de pagamentos?
SELECT
    op.payment_type,
    COUNT(op.order_id) AS pedidos,
    AVG(op.payment_value) AS AVG_pagamento,
    MAX(op.payment_installments) AS max_parcelas
FROM order_payments op
GROUP BY op.payment_type

```

	ABC payment_type	123 pedidos	123 AVG_pagamento	123 max_parcelas	
1	boleto	19,784	145.0344354023	1	
2	credit_card	76,795	163.3190206394	24	
3	debit_card	1,529	142.5701700458	1	
4	not_defined	3	0	1	
5	voucher	5,775	65.7033541126	1	

```

• -- Qual a valor mínimo, máximo, médio e as soma total paga por cada tipo de pagamento e número de parcelas disponíveis?
SELECT
    op.payment_type,
    op.payment_installments,
    MIN(op.payment_value) AS valor_minimo,
    MAX(op.payment_value) AS valor_maximo,
    AVG(op.payment_value) AS valor_medio,
    SUM(op.payment_value) AS soma_valor
FROM order_payments op
GROUP BY op.payment_type, op.payment_installments

```

	696 payment_type	123 payment_installments	123 valor_minimo	123 valor_maximo	123 valor_medio	123 soma_valor
1	boleto	1	11.62	7,274.88	145.0344354023	2,869,361.26999999753
2	credit_card	0	58.69	129.94	94.315	188.63
3	credit_card	1	0.01	13,664.08	95.8729298763	2,440,445.42999999927
4	credit_card	2	20.03	2,442.82	127.2281503263	1,579,283.03000000054
5	credit_card	3	30	3,256.14	142.5393174649	1,491,103.8000000002
6	credit_card	4	40	3,126.5	163.9768399549	1,163,907.60999999966
7	credit_card	5	49.99	4,809.44	183.4652223707	961,174.30000000021
8	credit_card	6	59.99	2,828.42	209.8499515306	822,611.8100000001
9	credit_card	7	69.97	2,094.44	187.6736715867	305,157.38999999999
10	credit_card	8	37.58	6,929.31	307.7374273664	1,313,423.34000000038
11	credit_card	9	90.11	2,116.01	203.4408695652	131,015.91999999999
12	credit_card	10	37.67	4,950.34	415.0858370871	2,211,577.3400000001
13	credit_card	11	110.3	198.83	124.932173913	2,873.44
14	credit_card	12	121.09	1,543.25	321.6784962406	42,783.24
15	credit_card	13	130	278	150.4625	2,407.4
16	credit_card	14	139.98	417.45	167.9626666667	2,519.44
17	credit_card	15	151.93	2,647.76	445.5531081081	32,970.93
18	credit_card	16	160.12	651.05	292.694	1,463.47
19	credit_card	17	170	177.99	174.6025	1,396.82
20	credit_card	18	180.76	1,232.85	486.4833333333	13,135.05
21	credit_card	20	200.08	2,022.33	615.8017647059	10,468.63
22	credit_card	21	212.03	303.02	243.7	731.1
23	credit_card	22	228.71	228.71	228.71	228.71
24	credit_card	23	236.48	236.48	236.48	236.48
25	credit_card	24	241.42	1,440.1	610.0488888889	10,980.88
26	debit_card	1	13.38	4,445.5	142.5701700458	217,989.79000000001
27	not_defined	1	0	0	0	0
28	voucher	1	0	3,184.34	65.7033541126	379,436.87000000007

☰ -- Quantos produtos estão cadastrados na empresa por categoria?

```

SELECT
    p.product_category_name AS categoria,
    COUNT(p.product_id) AS qtd_produtos
FROM products p
GROUP BY p.product_category_name
ORDER BY qtd_produtos DESC

```

	categoria	qtd_produtos	
1	cama_mesa_banho	3,029	
2	esporte_lazer	2,867	
3	moveis_decoracao	2,657	
4	beleza_saude	2,444	
5	utilidades_domestic	2,335	
6	automotivo	1,900	
7	informatica_acessorio	1,639	
8	brinquedos	1,411	
9	relogios_presentes	1,329	
10	telefonica	1,134	
11	bebes	919	
12	perfumaria	868	
13	papelaria	849	
14	fashion_bolsas_e_ac	849	
15	cool_stuff	789	
16	ferramentas_jardim	753	
17	pet_shop	719	
18	[NULL]	610	
19	eletronicos	517	
20	construcao_ferrame	400	
21	eletrodomesticos	370	
22	malas_acessorios	349	
23	consoles_games	317	
24	moveis_escritorio	309	
25	instrumentos_music	289	
26	eletroportateis	231	
27	casa_construcao	225	
28	livros_interesse_gera	216	
29	fashion_calcados	173	
30	moveis_sala	156	

```

-- Qual o número total de pedidos únicos, a data mínima e máxima de envio, o valor máximo, mínimo e médio do frete
-- dos pedidos abaixo de R$ 1.100 por cada vendedor? */
SELECT
    oi.seller_id,
    COUNT(DISTINCT oi.order_id) AS pedidos,
    MIN(DATE(oi.shipping_limit_date)) AS data_min,
    MAX(DATE(oi.shipping_limit_date)) AS data_max,
    MAX(oi.freight_value) AS frete_max,
    MIN(oi.freight_value) AS frete_min,
    AVG(oi.freight_value) AS frete_medio
FROM order_items oi
WHERE oi.price < 1100
GROUP BY oi.seller_id

```


	ABC seller_id	123 pedidos	ABC data_min	ABC data_max	123 frete_max	123 frete_min	123 frete_medio
1	0015a82c2db000af6aaaf3ae2ecb0532	3	2017-10-12	2017-10-24	21.02	21.02	21.02
2	001cca7ae9ae17fb1caed9dfb1094831	200	2017-02-08	2018-07-19	114.62	14.72	37.0466108787
3	001e6ad469a905060d959994f1b41e4f	1	2018-08-13	2018-08-13	17.94	17.94	17.94
4	002100f778ceb8431b7a1020ff7ab48f	51	2017-09-20	2018-04-18	34.15	4.91	14.4301818182
5	003554e2dce176b5555353e4f3555ac8	1	2017-12-21	2017-12-21	19.38	19.38	19.38
6	004c9cd9d87a3c30c522c48c4fc07416	158	2017-02-02	2018-05-10	133.73	7.67	20.8895882353
7	00720abe85ba0859807595bbf045a33b	13	2018-02-01	2018-08-07	38.6	6.1	12.1530769231
8	00ab3eff1b5192e5f1a63bcecfce11c8	1	2017-03-23	2017-03-23	12.08	12.08	12.08
9	00d8b143d12632bad99c0ad66ad52825	1	2018-03-27	2018-03-27	51.1	51.1	51.1
10	00ee68308b45bc5e2660cd833c3f81cc	135	2017-09-25	2018-08-15	98.02	2.34	18.4922093023

```

/* Quantos clientes únicos tiveram seu pedidos com status de “processing”,
“shipped” e “delivered”, feitos entre os dias 01 e 31 de Outubro de 2016. Mostrar
o resultado somente se o número total de clientes for acima de 5. */
SELECT
    o.order_status,
    COUNT(DISTINCT o.customer_id) AS clientes_unicos
FROM orders o
WHERE o.order_status IN ('processing', 'shipped', 'delivered')
    AND o.order_purchase_timestamp BETWEEN '2016-10-01' AND '2016-10-31'
GROUP BY o.order_status
HAVING COUNT(DISTINCT o.customer_id) > 5

```

	ABC order_status	123 clientes_unicos
1	delivered	265
2	shipped	8

```

-- Quais são os top 10 vendedores com mais clientes?
SELECT
    s.seller_id,
    COUNT(c.customer_id) AS qtde_clientes
FROM orders o LEFT JOIN order_items oi ON oi.order_id = o.order_id
    LEFT JOIN sellers s ON s.seller_id = oi.seller_id
    LEFT JOIN customer c ON c.customer_id = o.customer_id
GROUP BY s.seller_id
ORDER BY qtde_clientes DESC
LIMIT 10

```

	ABC seller_id	123 qtde_clientes
1	6560211a19b47992c3666cc44a7e94c0	2,033
2	4a3ca9315b744ce9f8e9374361493884	1,987
3	1f50f920176fa81dab994f9023523100	1,931
4	cc419e0650a3c5ba77189a1882b7556a	1,775
5	da8622b14eb17ae2831f4ac5b9dab84a	1,551
6	955fee9216a65b617aa5c0531780ce60	1,499
7	1025f0e2d44d7041d6cf58b6550e0bfa	1,428
8	7c67e1448b00f6e969d365cea6b010ab	1,364
9	ea8482cd71df3c1969d7b9473ff13abc	1,203
10	7a67c85e85bb2ce8582c35f2203ad736	1,171

```

-- Qual o número de pedidos com o tipo de pagamento igual a “boleto”?
SELECT
    COUNT(o.order_id)
FROM orders o
WHERE o.order_id IN (SELECT op.order_id
                     FROM order_payments op
                     WHERE op.payment_type = 'boleto')

```

	123 COUNT(o.order_id)
1	19,784

```

/* Quantos clientes avaliaram o pedido com 5 estrelas? */
SELECT
    COUNT(o.customer_id) AS qtde_clientes
FROM orders o LEFT JOIN order_reviews or2 ON or2.order_id = o.order_id
WHERE or2.review_score = 5

```

	123 qtde_clientes
1	57,420

```

/* Quantos clientes avaliaram o pedido com 1
estrelas? */
SELECT
    COUNT(o.customer_id) AS qtde_clientes
FROM orders o LEFT JOIN order_reviews or2 ON or2.order_id = o.order_id
WHERE or2.review_score = 1

```

	123 qtde_clientes
1	11,858

```

/* Crie uma tabela que mostre a média de avaliações por dia, a média de preço por dia, a
soma dos preços por dia, o preço mínimo por dia, o número de pedidos por dia e o número de
clientes únicos que compraram no dia. */
SELECT
    t1.date_,
    t1.avg_review,
    t2.avg_price,
    t2.sum_price,
    t2.min_price,
    t3.pedido_por_dia,
    t3.clientes_unicos
FROM (SELECT
        DATE( review_creation_date ) AS date_,
        AVG( review_score ) AS avg_review
    FROM order_reviews or2
    GROUP BY DATE( review_creation_date )) AS t1 LEFT JOIN ( SELECT
        DATE( oi.shipping_limit_date ) AS date_,
        AVG( price ) AS avg_price,
        SUM( price ) AS sum_price,
        MIN( price ) AS min_price
    FROM order_items oi
    GROUP BY DATE( oi.shipping_limit_date )) AS t2 ON ( t2.date_ = t1.date_)
LEFT JOIN (SELECT
        DATE( o.order_purchase_timestamp ) AS date_,
        COUNT( o.order_id ) AS pedido_por_dia,
        COUNT( DISTINCT o.customer_id ) AS clientes_unicos
    FROM orders o
    GROUP BY DATE( o.order_purchase_timestamp )) AS t3 ON ( t3.date_ = t1.date_)

```

	<small>ABC</small> date_	<small>123</small> avg_review	<small>123</small> avg_price	<small>123</small> sum_price	<small>123</small> min_price	<small>123</small> pedido_por_dia	<small>123</small> clientes_unicos
55	2017-01-12	5	106.5966666667	319.79	10.99	13	13
56	2017-01-13	5	114.8257142857	803.78	7.9	12	12
57	2017-01-14	4.2	249.97125	1,999.77	48.9	18	18
58	2017-01-17	4.8	243.568	2,435.68	54.9	32	32
59	2017-01-18	4.875	60.1842857143	842.58	10.99	33	33
60	2017-01-19	4.5	184.6438461538	2,400.37	10.49	29	29
61	2017-01-20	5	167.185	3,343.7	10.99	29	29
62	2017-01-21	4.3636363636	78.2034210526	2,971.73	8.9	24	24
63	2017-01-22	5	109.3463157895	4,155.16	10.99	31	31
64	2017-01-24	4.2857142857	110.15875	3,525.08	10.3	40	40
65	2017-01-25	3.7857142857	66.735	2,402.46	9.9	63	63

```

/* Crie uma consulta que exiba o código do produto e a categoria de cada produto
com base no seu preço:
Preço abaixo de 50 → Categoria A
Preço entre 50 e 100 → Categoria B
Preço entre 100 e 500 → Categoria C
Preço entre 500 e 1500 → Categoria D
Preço acima de 1500 → Categoria E
*/

```

```

SELECT
    oi.product_id,
    oi.price,
    CASE
        WHEN oi.price < 50.0 THEN 'categoria a'
        WHEN oi.price < 100.0 THEN 'categoria b'
        WHEN oi.price < 500.0 THEN 'categoria c'
        WHEN oi.price < 1500.0 THEN 'categoria d'
        ELSE 'categoria e'
    END AS categoria
FROM order_items oi

```

	<small>ABC</small> product_id	<small>123</small> price	<small>ABC</small> categoria
1	4244733e06e7ecb4970a6e2683c13e61	58.9	categoria b
2	e5f2d52b802189ee658865ca93d83a8f	239.9	categoria c
3	c777355d18b72b67abbef9df44fd0fd	199	categoria c
4	7634da152a4610f1595efa32f14722fc	12.99	categoria a
5	ac6c3623068f30de03045865e4e10089	199.9	categoria c
6	ef92defde845ab8450f9d70c526ef70f	21.9	categoria a
7	8d4f2bb7e93e6710a28f34fa83ee7d28	19.9	categoria a
8	557d850972a7d6f792fd18ae1400d9b6	810	categoria d
9	310ae3c140ff94b03219ad0adc3c778f	145.95	categoria c
10	4535b0e1091c278dfd193e5a1d63b39f	53.99	categoria b
11	d63c1011f49d98b976c352955b1c4bea	59.99	categoria b

```

-- Calcule a quantidade de produtos para cada uma das categorias criadas anteriormente
SELECT
    COUNT(oi.product_id),
    CASE
        WHEN oi.price < 50.0 THEN 'categoria a'
        WHEN oi.price >= 50.0 AND oi.price < 100.0 THEN 'categoria b'
        WHEN oi.price >= 100.0 AND oi.price < 500.0 THEN 'categoria c'
        WHEN oi.price >= 500.0 AND oi.price < 1500.0 THEN 'categoria d'
        ELSE 'categoria e'
    END AS categoria
FROM order_items oi
GROUP BY categoria

```

	123 COUNT(oi.product_id) ▾	ABC categoria
1	39,024	categoria a
2	33,141	categoria b
3	37,246	categoria c
4	2,902	categoria d
5	337	categoria e

```
● /* Crie uma consulta que exiba a data de compra, o valor de cada venda e o total acumulado de vendas até aquela data. */
SELECT
    DATE(o.order_purchase_timestamp),
    oi.price,
    SUM(oi.price) OVER (ORDER BY DATE(o.order_purchase_timestamp)) AS total_acumulado
FROM orders o LEFT JOIN order_items oi ON oi.order_id = o.order_id
WHERE oi.price IS NOT NULL
```

	DATE(o.order_purchase_timestamp) ▾	123 price ▾	123 total_acumulado ▾
1	2016-09-04	39.99	72.89
2	2016-09-04	32.9	72.89
3	2016-09-05	59.5	132.39
4	2016-09-15	44.99	267.36
5	2016-09-15	44.99	267.36
6	2016-09-15	44.99	267.36
7	2016-10-02	100	367.36
8	2016-10-03	29.9	830.84
9	2016-10-03	21.5	830.84
10	2016-10-03	36.49	830.84
11	2016-10-03	128.9	830.84

```
● /* Crie uma consulta que exiba o estado do cliente, a categoria, a quantidade de produtos vendidos e o percentual de vendas em relação ao total vendido no estado. */
SELECT
    estado,
    categoria,
    qtde_produtos,
    SUM(qtde_produtos) OVER (PARTITION BY estado) AS total_estado,
    (qtde_produtos*100.0/(SUM(qtde_produtos) OVER (PARTITION BY estado))) AS percentagem
FROM (SELECT
    c.customer_state AS estado,
    p.product_category_name AS categoria,
    COUNT(oi.product_id) AS qtde_produtos
    FROM orders o LEFT JOIN customer c ON c.customer_id = o.customer_id
    LEFT JOIN order_items oi ON oi.order_id = o.order_id
    LEFT JOIN products p ON oi.product_id = p.product_id
    GROUP BY estado, categoria
    HAVING categoria IS NOT NULL)
```

	ABC estado ▾	ABC categoria ▾	123 qtde_produtos ▾	123 total_estado ▾	123 percentagem ▾
1	AC	artigos_de_natal	1	90	1.1111111111
2	AC	automotivo	4	90	4.4444444444
3	AC	bebes	3	90	3.3333333333
4	AC	beleza_saude	7	90	7.7777777778
5	AC	brinquedos	3	90	3.3333333333
6	AC	cama_mesa_banho	4	90	4.4444444444
7	AC	consoles_games	1	90	1.1111111111
8	AC	cool_stuff	1	90	1.1111111111
9	AC	eletronicos	4	90	4.4444444444
10	AC	eletroportateis	1	90	1.1111111111
11	AC	esporte_lazer	9	90	10

```

-- Crie uma consulta SQL usando a cláusula WITH para calcular o total de vendas para cada categoria e exiba o resultado.
WITH vendas AS (
    SELECT
        p.product_category_name AS categoria,
        SUM(oi.price) AS total_vendas
    FROM order_items oi LEFT JOIN products p ON p.product_id = oi.product_id
    WHERE p.product_category_name IS NOT NULL
    GROUP BY p.product_category_name
)
SELECT
    categoria,
    total_vendas
FROM vendas

```

	categoria	total_vendas
1	agro_industria_e_comercio	72,530.47
2	alimentos	29,393.4100000002
3	alimentos_bebidas	15,179.48
4	artes	24,202.64
5	artes_e_artesanato	1,814.01
6	artigos_de_festas	4,485.18
7	artigos_de_natal	8,800.82
8	audio	50,688.5
9	automotivo	592,720.110000013
10	bebes	411,764.8900000016
11	bebidas	22,428.7
12	beleza_saude	1,258,681.339999968
13	brinquedos	483,946.6000000026
14	cama_mesa_banho	1,036,988.6800000716
15	casa_conforto	58,572.0400000003
16	casa_conforto_2	760.27
17	casa_construcao	83,088.1199999998
18	cds_dvds_musicais	730
19	cine_foto	6,933.46
20	climatizacao	55,024.9600000001
21	consoles_games	157,465.2199999991
22	construcao_ferramentas_co	144,677.5899999998
23	construcao_ferramentas_fer	15,903.95
24	construcao_ferramentas_ilu	41,080
25	construcao_ferramentas_jar	25,715.89
26	construcao_ferramentas_se	40,544.5200000001
27	cool_stuff	635,290.8500000037
28	dvds_blu_ray	5,999.39
29	eletrodomesticos	80,171.53

```

-- Crie uma consulta usando a cláusula WITH para calcular a receita total por mês e exiba o mês com a maior receita.
WITH receita AS (
    SELECT
        STRFTIME('%m', o.order_purchase_timestamp) AS mes,
        SUM(oi.price) AS vendas
    FROM orders o LEFT JOIN order_items oi ON oi.order_id = o.order_id
    GROUP BY STRFTIME('%m', o.order_purchase_timestamp)
)
SELECT
    mes,
    vendas
FROM receita
WHERE vendas = (SELECT MAX(vendas) FROM receita)

```

	mes	vendas
1	05	1,502,588.8199998986

```

/* Crie uma consulta SQL com as seguintes colunas: 1) Categoria, 2)Preço, 3)Date limite de envio, 4) Primeira compra, 5) Número de dias que o produto foi comprado a partir da primeira compra dentro da mesma categoria. */
WITH tabela_temp AS (
    SELECT
        p.product_category_name AS categoria,
        oi.order_id AS compra,
        oi.price AS preco,
        oi.shipping_limit_date AS data_limite_envio,
        FIRST_VALUE(oi.shipping_limit_date) OVER (PARTITION BY p.product_category_name ORDER BY oi.shipping_limit_date ASC) AS primeira_compra
    FROM order_items oi LEFT JOIN products p ON p.product_id = oi.product_id
    WHERE p.product_category_name IS NOT NULL
)
SELECT
    categoria,
    preco,
    STRFTIME('%Y-%m-%d', data_limite_envio) AS data_limite_envio,
    STRFTIME('%Y-%m-%d', primeira_compra) AS primeira_compra,
    CAST((JULIANDAY(data_limite_envio) - JULIANDAY(primeira_compra)) AS INTEGER) AS dias
FROM tabela_temp

```

	ABC categoria	123 preco	ABC data_limite_envio	ABC primeira_compra	123 dias	
1	agro_industria_e_comercio	21.99	2017-01-27	2017-01-27	0	
2	agro_industria_e_comercio	21.99	2017-01-27	2017-01-27	0	
3	agro_industria_e_comercio	21.99	2017-02-04	2017-01-27	8	
4	agro_industria_e_comercio	21.99	2017-02-09	2017-01-27	13	
5	agro_industria_e_comercio	21.99	2017-02-12	2017-01-27	15	
6	agro_industria_e_comercio	21.99	2017-02-17	2017-01-27	21	
7	agro_industria_e_comercio	21.99	2017-02-20	2017-01-27	24	
8	agro_industria_e_comercio	92.9	2017-02-21	2017-01-27	25	
9	agro_industria_e_comercio	21.99	2017-02-22	2017-01-27	26	
10	agro_industria_e_comercio	21.99	2017-02-25	2017-01-27	29	
11	agro_industria_e_comercio	59.99	2017-03-23	2017-01-27	55	
12	agro_industria_e_comercio	22	2017-03-24	2017-01-27	56	
13	agro_industria_e_comercio	59.99	2017-05-11	2017-01-27	104	
14	agro_industria_e_comercio	589.99	2017-05-15	2017-01-27	107	
15	agro_industria_e_comercio	869.97	2017-05-25	2017-01-27	117	
16	agro_industria_e_comercio	59.99	2017-06-01	2017-01-27	125	
17	agro_industria_e_comercio	1,390	2017-07-06	2017-01-27	160	
18	agro_industria_e_comercio	1,180	2017-07-20	2017-01-27	174	

```

/* Qual a categoria que possui o produto com o maior número de dias entre a primeira compra da categoria e a sua data limite de entrega? */
WITH table_temp AS (
    SELECT
        p.product_category_name AS categoria,
        oi.shipping_limit_date AS limite_entrega,
        FIRST_VALUE(o.order_purchase_timestamp) OVER (PARTITION BY p.product_category_name ORDER BY o.order_purchase_timestamp ASC) AS primeira_compra
    FROM orders o LEFT JOIN order_items oi ON oi.order_id = o.order_id
    LEFT JOIN products p ON p.product_id = oi.product_id
    WHERE categoria IS NOT NULL
)
SELECT
    categoria,
    STRFTIME('%Y-%m-%d', limite_entrega) AS limite_entrega,
    STRFTIME('%Y-%m-%d', primeira_compra) AS primeira_compra,
    MAX(CAST((JULIANDAY(limite_entrega) - JULIANDAY(primeira_compra)) AS INTEGER)) AS dias_max
FROM table_temp

```

	ABC categoria	ABC limite_entrega	ABC primeira_compra	123 dias_max	
1	utilidades_domesticas	2020-04-09	2016-10-04	1,283	