

4. (30 points) Import your chosen dataset.

a. Check that the data structures were imported correctly; if necessary, fix them manually (data types, column names, remove unnecessary columns).

b. Perform basic checks and queries:

SELECT * FROM table LIMIT 10;

Query

Query History

Scratch Pad X

1

-- 1. this previews first 10 rows from orders

2

SELECT * FROM olist_orders LIMIT 10;

3

4

-- 2. this gets delivered orders after 2018-01-01 sorted by delivery date

5

SELECT order_id, customer_id, order_status, order_delivered_customer_date

6

FROM olist_orders

7

WHERE order_status = 'delivered' AND order_delivered_customer_date > '2018-01-01'

8

ORDER BY order_delivered_customer_date ASC;

9

10

-- 3. aggregation: count orders per status

11

SELECT order_status, COUNT(*) AS total_orders

12

FROM olist_orders

13

GROUP BY order_status;

14

15

Data Output

Messages

Notifications

Showing rows: 1 to 10

Page No: 1

of 1

	order_id [PK] text	customer_id text	order_status text	order_purchase_timestamp timestamp without time zone	order_approved_at timestamp without time zone	order_delivered_carrier_date timestamp without time zone	order_delivered_c timestamp withou
1	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a92...	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-04 19:55:00	2017-10-10 21:25
2	53cdb2fc8bc7dce0b6741e21502734...	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2018-07-24 20:41:37	2018-07-26 03:24:27	2018-07-26 14:31:00	2018-08-07 15:27
3	47770eb9100c2d0c44946d9cf07ec6...	41ce2a54c0b03bf3443c3d931a367089	delivered	2018-08-08 08:38:49	2018-08-08 08:55:23	2018-08-08 13:50:00	2018-08-17 18:06
4	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbe7375364d82	delivered	2017-11-18 19:28:06	2017-11-18 19:45:59	2017-11-22 13:39:59	2017-12-02 00:28
5	ad21c59c0840e6cb83a9ceb5573f81...	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2018-02-13 21:18:39	2018-02-13 22:20:29	2018-02-14 19:46:34	2018-02-16 18:17
6	a4591c265e18cb1dcee52889e2d8ac...	503740e9ca751ccdda7ba28e9ab8f608	delivered	2017-07-09 21:57:05	2017-07-09 22:10:13	2017-07-11 14:58:04	2017-07-26 10:57
7	136cce7faa42fdb2cef53fcd79a6098	ed0271e0b7da060a393796590e7b73...	invoiced	2017-04-11 12:22:08	2017-04-13 13:25:17	[null]	[null]
8	6514b8ad8028e9f2cc2374ded24578f3	9bdf08b4b3b52b5526ff42d37d47f222	delivered	2017-05-16 13:10:30	2017-05-16 13:22:11	2017-05-22 10:07:46	2017-05-26 12:55

a query with filtering (WHERE) and sorting (ORDER BY);

5

-- 2. this gets delivered orders after 2018-01-01 sorted by delivery date

6

SELECT order_id, customer_id, order_status, order_delivered_customer_date

7

FROM olist_orders

8

WHERE order_status = 'delivered' AND order_delivered_customer_date > '2018-01-01'

9

ORDER BY order_delivered_customer_date ASC;

10

-- 3. aggregation: count orders per status

11

SELECT order_status, COUNT(*) AS total_orders

12

FROM olist_orders

13

GROUP BY order_status;

14

15

Data Output

Messages

Notifications

aggregation (GROUP BY) with functions COUNT , AVG , MIN , MAX ;

```

11 -- 3. aggregation: count orders per status
12 SELECT order_status, COUNT(*) AS total_orders
13 FROM olist_orders
14 GROUP BY order_status;
15
16 -- 4. aggregation with AVG: average payment value per payment type
17 SELECT payment_type, AVG(payment_value) AS avg_payment
18 FROM olist_order_payments
19 GROUP BY payment_type;
20
21 -- 5. aggregation with MIN/MAX: Earliest and latest purchase timestamp per status
22 SELECT MIN(order_purchase_timestamp) AS first_order,
23        MAX(order_purchase_timestamp) AS last_order

```

Data Output Messages Notifications

	order_status text	total_orders bigint
1	shipped	1107
2	unavailable	609
3	invoiced	314
4	created	5
5	approved	2
6	processing	301
7	delivered	96478
8	canceled	625

```

21 -- 5. aggregation with MIN/MAX: Earliest and latest purchase date
22 SELECT MIN(order_purchase_timestamp) AS first_order,
23        MAX(order_purchase_timestamp) AS last_order
24 FROM olist_orders;
25
26 -- 6. JOIN example: Join orders with customers to get customer city/state

```

Data Output Messages Notifications

SQL

	first_order timestamp without time zone	last_order timestamp without time zone
1	2016-09-04 21:15:19	2018-10-17 17:30:18

```

15
16 -- 4. aggregation with AVG: average payment value per payment type
17 SELECT payment_type, AVG(payment_value) AS avg_payment
18 FROM olist_order_payments
19 GROUP BY payment_type;
20
21 -- 5. aggregation with MIN/MAX: Earliest and latest purchase date
22 SELECT MIN(order_purchase_timestamp) AS first_order,
23        MAX(order_purchase_timestamp) AS last_order
24 FROM olist_orders;
25
26 -- 6. JOIN example: Join orders with customers to get customer city/state

```

Data Output Messages Notifications

SQL

	payment_type character varying (50)	avg_payment numeric
1	not_defined	0.00000000000000000000
2	boleto	145.0344354023453296
3	debit_card	142.5701700457815566
4	voucher	65.7033541125541126
5	credit_card	163.3190206393645420

at least one JOIN between tables.

```

26 -- 6. JOIN example: Join orders with customers to get customer city/state
27 SELECT o.order_id, c.customer_city, c.customer_state, o.order_status
28 FROM olist_orders o
29 JOIN olist_customers c ON o.customer_id = c.customer_id
30 LIMIT 10;
31
32 -- =====
33 -- c. Define 10 analytical topics for your project
34 -- =====
35

```

Data Output Messages Notifications

	order_id text	customer_city text	customer_state text	order_status text
1	e481f51cbdc54678b7cc49136f2d6af7	sao paulo	SP	delivered
2	53cdb2fc8bc7dce0b6741e21502734...	barreiras	BA	delivered
3	47770eb9100c2d0c44946d9cf07ec6...	vianopolis	GO	delivered
4	949d5b44dbf5de918fe9c16f97b45f8a	sao goncalo do amarante	RN	delivered
5	ad21c59c0840e6cb83a9ceb5573f81...	santo andre	SP	delivered
6	a4591c265e18cb1dcee52889e2d8ac...	congonginhas	PR	delivered
7	136cce7faa42fdb2cefd53fdc79a6098	santa rosa	RS	invoiced
8	6514b8ad8028c9f2cc2374ded245783f	nilopolis	RJ	delivered
9	76c6e866289321a7c93b82b54852dc...	faxinalzinho	RS	delivered

c. Define 10 analytical topics for your project (e.g., average scores per subject, number of admissions by year/department, sales by region, etc.). Write an SQL query for each topic and make sure the results are correct. Save everything in a queries.sql file with short comments (explaining what each query does).

```

37 SELECT DATE_PART('year', order_purchase_timestamp) AS year,
38        COUNT(*) AS total_orders
39 FROM olist_orders
40 GROUP BY year
41 ORDER BY year;
42
43 -- 2. Average review score per product
44 SELECT oi.product_id, AVG(r.review_score) AS avg_score
45 FROM olist_order_items oi

```

Data Output Messages Notifications

	year double precision	total_orders bigint
1	2016	329
2	2017	45101
3	2018	54011

```

44 SELECT oi.product_id, AVG(r.review_score) AS avg_score
45 FROM olist_order_items oi
46 JOIN olist_order_reviews r ON oi.order_id = r.order_id
47 GROUP BY oi.product_id
48 ORDER BY avg_score DESC;
49
50 -- 3. Top 10 states by number of customers
51 SELECT customer_state, COUNT(*) AS total_customers
52 FROM olist_customers

```

Data Output Messages Notifications

	product_id character varying (50)	avg_score numeric
1	ec87fe7c217a713732b71344bc43274b	5.0000000000000000
2	b861b1b731a9102704784b9912aca8...	5.0000000000000000
3	96ee9343533c399d245845830908f073	5.0000000000000000
4	9ea4c0cee3567625c3b2dcff46ef152a	5.0000000000000000
5	69fb24f0cd077f460768e66b89c3565e	5.0000000000000000
6	685c297d0eda4eb885028d8699df7a8b	5.0000000000000000
7	1684f99c7e0e67403e52a7783366870f	5.0000000000000000
8	eb2e510c7e0f62651d0156e769928970	5.0000000000000000
9

```

50 -- 3. Top 10 states by number of customers
51 SELECT customer_state, COUNT(*) AS total_customers
52 FROM olist_customers
53 GROUP BY customer_state
54 ORDER BY total_customers DESC
55 LIMIT 10;
56
57 -- 4. Average payment value per installment count
58 SELECT payment_installments, AVG(payment_value) AS avg_
59 FROM olist_order_payments
60 GROUP BY payment_installments

```

Data Output Messages Notifications

	customer_state text	total_customers bigint
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033

```

57 -- 4. Average payment value per installment count
58 SELECT payment_installments, AVG(payment_value) AS avg_payment
59 FROM olist_order_payments
60 GROUP BY payment_installments
61 ORDER BY payment_installments;
62
63 -- 5. Number of orders by delivery status (delivered, shipped, canceled)
64 SELECT order_status, COUNT(*) AS total
65 FROM olist_orders

```

Data Output Messages Notifications

	payment_installments integer	avg_payment numeric
1	0	94.3150000000000000
2	1	112.4202291325695581
3	2	127.2281503262708451
4	3	142.5393174648695153
5	4	163.9768399549168780
6	5	183.4652223706814278
7	6	209.8499515306122449
8	7	187.6736715867158672

```

63 -- 5. Number of orders by delivery status (delivered)
64 SELECT order_status, COUNT(*) AS total
65 FROM olist_orders
66 GROUP BY order_status
67 ORDER BY total DESC;
68
69 -- 6. Average delivery time (customer date - purchase date)
70 SELECT AVG(order_delivered_customer_date - order_purchase_timestamp) AS avg_delivery_days
71 FROM olist_orders
72 WHERE order_status = 'delivered';
73
74 -- 7. Top 10 most sold products
75 SELECT oi.product_id, COUNT(*) AS total_sold

```

Data Output Messages Notifications

	order_status text	total bigint
1	delivered	96478
2	shipped	1107
3	canceled	625
4	unavailable	609
5	invoiced	314
6	processing	301
7	created	5
8	approved	2

```

69 -- 6. Average delivery time (customer date - purchase date)
70 SELECT AVG(order_delivered_customer_date - order_purchase_timestamp) AS avg_delivery_days
71 FROM olist_orders
72 WHERE order_status = 'delivered';
73
74 -- 7. Top 10 most sold products
75 SELECT oi.product_id, COUNT(*) AS total_sold
76 FROM olist_order_items oi
77 GROUP BY oi.product_id
78 ORDER BY total_sold DESC
79 LIMIT 10;

```

Data Output Messages Notifications

	avg_delivery_days interval
1	12 days 13:23:49.957272

Showing rows: 1


```

74 -- 7. Top 10 most sold products
75 SELECT oi.product_id, COUNT(*) AS total_sold
76 FROM olist_order_items oi
77 GROUP BY oi.product_id
78 ORDER BY total_sold DESC
79 LIMIT 10;
80
81 -- 8. Seller performance: total sales value per seller
82 SELECT oi.seller_id, SUM(oi.price) AS total_sales
83 FROM olist_order_items oi
84 GROUP BY oi.seller_id
85 ORDER BY total_sales DESC
86 LIMIT 10;

```

Data Output Messages Notifications

	product_id character varying (50)	total_sold bigint
1	aca2eb7d00ea1a7b8ebd4e68314663af	527
2	99a4788cb24856965c36a24e339b60...	488
3	422879e10f46682990de24d770e7f83d	484
4	389d119b48cf3043d311335e499d9c6b	392
5	368c6c730842d78016ad823897a372...	388
6	53759a2ecddad2bb87a079a1f1519f73	373
7	d1c427060a0f73f6b889a5c7c61f2ac4	343
8	53b36df67ebb7c41585e8d54d6772e08	323
9	154e7e31ebfa092203795c972e5804a6	281
Total rows: 10		Query complete 00:00:00.139

```

82 SELECT oi.seller_id, SUM(oi.price) AS total_sales
83 FROM olist_order_items oi
84 GROUP BY oi.seller_id
85 ORDER BY total_sales DESC
86 LIMIT 10;
87
88 -- 9. Average freight value by state
89 SELECT c.customer_state, AVG(oi.freight_value) AS avg_fre
90 FROM olist_order_items oi
91 JOIN olist_orders o ON oi.order_id = o.order_id
92 JOIN olist_customers c ON o.customer_id = c.customer_id

```

Data Output Messages Notifications

	seller_id character varying (50)	total_sales numeric
1	4869f7a5dfa277a7dca6462dcf3b52b2	229472.63
2	53243585a1d6dc2643021fd1853d89...	222776.05
3	4a3ca9315b744ce9f8e93743614938...	200472.92
4	fa1c13f2614d7b5c4749cbc52fecda94	194042.03
5	7c67e1448b00f6e969d365cea6b010...	187923.89
6	7e93a43ef30c4f03f38b393420bc753a	176431.87
7	da8622b14eb17ae2831f4ac5b9dab8...	160236.57
8	7a67c85e85bb2ce8582c35f2203ad7...	141745.53
9	1025f0e2d44d7041d6cf58b6550e0bfa	138968.55

```

89 SELECT c.customer_state, AVG(oi.freight_value) AS avg_freight
90 FROM olist_order_items oi
91 JOIN olist_orders o ON oi.order_id = o.order_id
92 JOIN olist_customers c ON o.customer_id = c.customer_id
93 GROUP BY c.customer_state
94 ORDER BY avg_freight DESC;
95
96 -- 10. Number of reviews per score (distribution of ratings)
97 SELECT review_score, COUNT(*) AS review_count
98 FROM olist_order_reviews
99 GROUP BY review_score
100 ORDER BY review_score;
101

```

Data Output Messages Notifications



	customer_state text	avg_freight numeric
1	RR	42.9844230769230769
2	PB	42.7238039867109635
3	RO	41.0697122302158273
4	AC	40.0733695652173913
5	PI	39.1479704797047970
6	MA	38.2570024271844660
7	TO	37.2466031746031746
8	SE	36.6531688311688312
9	AI	35.8436711711711712

```

96 -- 10. Number of reviews per score (distribution of ratings)
97 SELECT review_score, COUNT(*) AS review_count
98 FROM olist_order_reviews
99 GROUP BY review_score
100 ORDER BY review_score;
101

```

Data Output Messages Notifications



	review_score integer	review_count bigint
1	1	11424
2	2	3151
3	3	8179
4	4	19142
5	5	57328