E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Category 1: CMO (Chief Marketing Officer) Analysis

Goal:

To measure the effectiveness of marketing channels, understand true customer value, and track loyalty and retention. These queries analyze acquisition, customer lifetime value (CLV), and cohort behavior to guide strategic marketing spend.

Query 1.1: Channel Acquisition Summary

Description:

This view provides a high-level summary of acquisition performance by traffic source. It counts the total unique sessions, the total orders generated (by counting unique sessions with a 'purchase' event), and segments this traffic into "Guest" vs. "Logged-in" sessions. It's the first step in identifying which channels drive high-volume traffic versus high-value, purchasing traffic.

```sql
-- CMO ANALYSIS ---------------------------
-- Acquisition Analysis -------------------
SELECT
    traffic_source,
    COUNT(DISTINCT session_id) AS total_sessions,
    COUNT(CASE
        WHEN event_type = 'purchase' THEN event_type
        ELSE NULL
    END) total_orders,
    COUNT(DISTINCT CASE
            WHEN user_id IS NOT NULL THEN session_id
        END) AS logged_in_customers,
    COUNT(DISTINCT CASE
            WHEN user_id IS NULL THEN session_id
        END) AS guest_sessions
FROM
    events
GROUP BY traffic_source;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

traffic_source	total_sessions	total_orders	logged_in_customers	guest_sessions
adwords	205010	54542	54542	150468
email	306313	81706	81706	224607
facebook	67933	18305	18305	49628
organic	34301	9124	9124	25177
youtube	68202	18082	18082	50120

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.2: Channel Quality (CVR)

Description:

This query calculates the precise Conversion Rate (CVR) for each traffic source. A low CVR from a specific source (**e.g.,** 'Facebook') may indicate low-intent traffic or misaligned ad expectations.

```
43      -- ------------ Channel Quality -----------------
44      -- CVR traffic source
45      SELECT
46          e.traffic_source,
47          COUNT(DISTINCT session_id) AS total_sessions,
48          COUNT(DISTINCT CASE
49                  WHEN event_type = 'purchase' THEN session_id
50              END) AS total_order_sessions,
51          round(CAST(COUNT(DISTINCT CASE
52                      WHEN event_type = 'purchase' THEN session_id
53                  END)
54              AS DECIMAL (18 , 4 )) / COUNT(DISTINCT session_id), 4) AS CVR
55      FROM
56          events AS e
57      GROUP BY e.traffic_source
58      ORDER BY total_sessions DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

traffic_source	total_sessions	total_order_sessions	CVR
email	306313	81706	0.2667
adwords	205010	54542	0.2660
youtube	68202	18082	0.2651
facebook	67933	18305	0.2695
organic	34301	9124	0.2660

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.3: Browser Quality (CVR)

Description:

This query calculates the CVR for each browser. This is a key diagnostic tool. A low CVR from a specific browser (**e.g.**, 'Safari') indicates a critical bug or poor user experience on that platform (**e.g.**, iPhones) that needs immediate attention from the CX and development teams.

```
61
62      -- CVR browser
63  *   SELECT
64          e.browser,
65          COUNT(DISTINCT session_id) AS total_sessions,
66          COUNT(DISTINCT CASE
67                  WHEN event_type = 'purchase' THEN session_id
68              END) AS total_order_sessions,
69          CAST(COUNT(DISTINCT CASE
70                      WHEN event_type = 'purchase' THEN session_id
71                  END)
72              AS DECIMAL (18 , 4 )) / COUNT(DISTINCT session_id) AS CVR
73      FROM
74          events AS e
75      GROUP BY e.browser
76      ORDER BY total_sessions DESC;
77
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

browser	total_sessions	total_order_sessions	CVR
chrome	341142	91377	0.26785620
firefox	136615	36457	0.26685942
safari	135817	35932	0.26456187
ie	34312	8957	0.26104570
other	33873	9036	0.26676114

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.4: Customer Lifetime Value (CLV)

Description:

This query calculates the total *profit* generated by each customer. It calculates the historical CLV for *each customer* by summing (sale_price - cost) for all their non-returned orders ('Complete' or 'Shipped'). This identifies the company's VIP customers.

```sql
-- CLV
WITH OrderProfit AS (
    SELECT
        o.user_id,
        oi.order_id,
        SUM(oi.sale_price) AS total_revenue_per_order,
        SUM(p.cost) AS total_cost_per_order,
        (SUM(oi.sale_price) - SUM(p.cost)) AS total_profit_per_order
    FROM
        order_items oi
    JOIN
        products p ON oi.product_id = p.id
    JOIN
        orders o ON oi.order_id = o.order_id
    WHERE
        o.user_id IS NOT NULL
        AND o.status in('Complete', 'Shipped')
    GROUP BY
        o.user_id, oi.order_id
)

SELECT
    user_id,
    COUNT(order_id) AS total_orders,
    round(SUM(total_revenue_per_order),2) AS total_lifetime_revenue,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

user_id	total_orders	total_lifetime_revenue	total_lifetime_cost	historical_clv
1	1	159.99	81.13	78.86
2	1	22	10.38	11.62
3	2	233.46	119.37	114.09
4	1	148	61.57	86.43
6	1	144.82	62.79	82.03

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.5: Channel CLV (First-Touch Attribution)

Description:

This query attributes CLV back to the acquiring channel. It uses a UserFirstTouch CTE to identify the *first-ever* traffic source for each customer. It then calculates the average_clv_per_customer for each channel. This answers the most important marketing question: "Which channels acquire our most profitable customers long-term?"

```
113     -- Channels CLV
114 •   WITH CustomerCLV AS (
115         SELECT
116             o.user_id,
117             (SUM(oi.sale_price) - SUM(p.cost)) AS historical_clv
118         FROM
119             order_items oi
120         JOIN
121             products p ON oi.product_id = p.id
122         JOIN
123             orders o ON oi.order_id = o.order_id
124         WHERE
125             o.user_id IS NOT NULL
126             AND o.status in('Complete', 'Shipped')
127         GROUP BY
128             o.user_id
129     ),
130     UserFirstTouch AS (
131         SELECT
132             user_id,
133             traffic_source
134         FROM (
135             SELECT
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

traffic_source	total_acquired_customers	average_clv_per_customer	total_clv_from_channel
email	23842	58.77	1401185.08
organic	2577	58.77	151456.28
facebook	5215	58.69	306051.82
adwords	15976	58.66	937082.3
youtube	5301	57.36	304073.52

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.6: New vs. Returning Customer Analysis

Description:

This view measures customer loyalty by comparing the purchasing power of new vs. returning customers. It uses ROW_NUMBER() to tag each order as a user's 1st purchase ("New Customer") or a subsequent purchase ("Returning Customer"). It then calculates the AOV and revenue share for each group to prove the value of retention (e.g., "Returning customers spend 20% more per order").

```
163     -- New vs. Returning
164 •   WITH OrderRevenue AS (
165         SELECT
166             o.order_id,
167             o.user_id,
168             o.created_at,
169             SUM(oi.sale_price) AS order_revenue
170         FROM
171             orders o
172         JOIN
173             order_items oi ON o.order_id = oi.order_id
174         WHERE
175             o.status in('Complete', 'Shipped')
176         GROUP BY
177             o.order_id, o.user_id, o.created_at
178     ),
179     TaggedOrders AS (
180         SELECT
181             order_id,
182             user_id,
183             order_revenue,
184             ROW_NUMBER() OVER(PARTITION BY user_id ORDER BY created_at ASC) AS order_rank
185         FROM
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_type	total_orders	total_revenue	AOV	revenue_percentage
New Customer	52911	4577136.59	86.51	76.64
Returning Customer	16020	1394868.62	87.07	23.36

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.7: 12-Month Retention Cohort Analysis

Description:

This is the most powerful retention metric. It groups customers into "cohorts" based on their first-ever purchase month. It then tracks what percentage of each cohort returns to make another purchase in Month 1, Month 2... up to Month 12. This query was critical in discovering that our business has a long-term (7-12 month) repurchase cycle, not a seasonal one.

```
215     -- Cohort Analysis
216   WITH ValidPurchases AS (
217       SELECT DISTINCT
218           o.user_id,
219           DATE_FORMAT(o.created_at, '%Y-%m-01') AS purchase_month
220       FROM
221           orders o
222       JOIN
223           order_items oi ON o.order_id = oi.order_id
224       WHERE
225           o.status <> 'returned'
226           AND o.user_id IS NOT NULL
227   ),
228   UserCohorts AS (
229       SELECT
230           user_id,
231           MIN(purchase_month) AS cohort_month
232       FROM
233           ValidPurchases
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

cohort_month	cohort_size	month_1_retention	month_2_retention	month_3_retention	month_4_retention	month_5_reti	month_6_retent	month_7_reter	month_8_rete	month_9_rete	month_10_r	month_11_re	month_12_retention
2019-01-01	19	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	5.26315780	0.00000000	0.00000000	0.00000000	0.00000000
2019-02-01	34	0.00000000	0.00000000	0.00000000	5.88235290	5.88235290	5.88235290	0.00000000	5.88235290	0.00000000	2.94117640	5.88235290	2.94117640
2019-03-01	75	1.33333330	0.00000000	1.33333330	1.33333330	1.33333330	1.33333330	2.66666660	1.33333330	1.33333330	5.33333330	1.33333330	0.00000000
2019-04-01	97	1.03092780	1.03092780	1.03092780	0.00000000	2.06185560	1.03092780	0.00000000	1.03092780	3.09278350	0.00000000	1.03092780	0.00000000
2019-05-01	144	0.69444440	0.69444440	2.08333330	1.38888880	1.38888880	1.38888880	1.38888880	2.08333330	0.00000000	0.69444440	0.69444440	0.00000000
2019-06-01	171	0.58479530	1.75438590	1.16959060	1.16959060	1.75438590	0.58479530	0.58479530	1.16959060	0.58479530	1.16959060	1.75438590	0.00000000
2019-07-01	194	2.57731950	1.03092780	1.54639170	0.00000000	1.03092780	4.12371130	1.54639170	1.03092780	1.54639170	2.06185560	2.57731950	1.54639170
2019-08-01	192	0.52083330	1.56250000	1.04166660	1.04166660	1.56250000	1.56250000	1.04166660	0.00000000	4.68750000	1.56250000	3.12500000	2.08333330
2019-09-01	280	1.42857140	2.14285710	2.14285710	2.50000000	1.07142850	1.42857140	2.14285710	1.42857140	1.07142850	1.42857140	3.57142850	1.07142850
2019-10-01	287	2.43902430	2.43902430	2.78745640	2.43902430	1.04529610	1.39372820	1.04529610	2.78745640	1.74216020	0.69686410	0.00000000	1.74216020

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.8: Average Order Value (AOV) Trend

Description:

This query tracks the health of the Average Order Value (AOV) over time (both monthly and yearly). This was used to identify the strategic shift in our business.

```sql
SELECT
    DATE_FORMAT(o.created_at, '%Y-%m') AS purchase_month_year,
    COUNT(DISTINCT o.order_id) AS total_orders,
    AVG(oi.order_revenue) AS average_order_value
FROM
    orders o
        JOIN
    (SELECT
        order_id, SUM(sale_price) AS order_revenue
    FROM
        order_items
    GROUP BY order_id) oi ON o.order_id = oi.order_id
WHERE
    o.status <> 'returned'
GROUP BY purchase_month_year
ORDER BY purchase_month_year;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

purchase_month_year	total_orders	average_order_value
2019-01	19	102.76789479506643
2019-02	34	107.87176488427555
2019-03	75	95.27760021527608
2019-04	98	100.627959592001 77
2019-05	146	89.94075331132706
2019-06	177	80.78412453064138
2019-07	201	85.09323383207938
2019-08	207	95.0164735155981
2019-09	291	89.14223368634883
2019-10	310	84.66503246907266
2019-11	320	80.96165630146861
2019-12	415	84.770747066405 89
2020-01	411	91.56107048222619

```sql
-- AOV over years
SELECT
    DATE_FORMAT(o.created_at, '%Y') AS purchase_month_year,
    COUNT(DISTINCT o.order_id) AS total_orders,
    AVG(oi.order_revenue) AS average_order_value
FROM
    orders o
        JOIN
    (SELECT
        order_id, SUM(sale_price) AS order_revenue
    FROM
        order_items
    GROUP BY order_id) oi ON o.order_id = oi.order_id
WHERE
    o.status <> 'returned'
GROUP BY purchase_month_year
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

purchase_month_year	total_orders	average_order_value
2019	2293	87.26734418833666
2020	7772	86.61021238514829
2021	15031	87.12830358782925
2022	26312	85.24134322560151
2023	53072	86.48417724191617

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.9: Cart Abandonment by Channel

Description:

This query analyzes "cart abandonment" by traffic source. It isolates sessions that successfully *reached* the cart and calculates the percentage of those sessions that *failed* to purchase. This helps the CMO identify channels (e.g., 'Facebook') that drive low-intent "window-shopping" traffic, or where promotional offers may be misleading.

```
313      -- ------------ C.Marketing Behavior -----------------
314
315      WITH SessionFacts AS (
316          SELECT
317              session_id,
318              MAX(traffic_source) AS traffic_source,
319              MAX(CASE
320                      WHEN event_type = 'cart' THEN 1
321                      ELSE 0
322                  END) AS reached_cart,
323              MAX(CASE
324                      WHEN event_type = 'purchase' THEN 1
325                      ELSE 0
326                  END) AS did_purchase
327          FROM
328              events
329          GROUP BY
330              session_id
331      )
332      SELECT
333          traffic_source,
334          COUNT(session_id) AS total_sessions_reached_cart,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

traffic_source	total_sessions_reached_cart	total_sessions_purchased	cart_conversion_rate_percent
youtube	43182	18082	41.87392890
adwords	130001	54542	41.95506180
email	193988	81706	42.11910010
facebook	43372	18305	42.20464810
organic	21603	9124	42.23487470

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 1.10: Top Regions by Revenue & AOV

Description:

This query identifies the most valuable geographic markets (country, state, and city) by ranking them based on total revenue and AOV. This provides a clear guide for regional marketing spend, logistics planning, and expansion strategy.

```
346     -- Top region by Revenue & AOV
347
348 *   WITH OrderRevenue AS (
349         SELECT
350             o.order_id,
351             o.user_id,
352             SUM(oi.sale_price) AS order_revenue
353         FROM
354             orders o
355         JOIN
356             order_items oi ON o.order_id = oi.order_id
357         WHERE
358            o.status <> 'returned'
359         GROUP BY
360             o.order_id, o.user_id
361     ),
362     UserOrders AS (
363         SELECT
364             u.country,
365             u.state,
366             u.city,
367             o.order_id,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	state	city	total_orders	total_revenue	AOV
South Korea	Seoul	Seoul	1673	149635.17	89.44
United States	New York	New York	954	80778.99	84.67
Japan	Tokyo	Tokyo	846	78160.29	92.39
Brasil	Bahia	Salvador	624	49679.8	79.62
United Kingdom	England	London	557	48132.81	86.41
Brasil	São Paulo	São Paulo	538	46970.97	87.31
South Korea	Busan	Busan	486	45384.85	93.38

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Category 2: CX (Customer Experience) Analysis

Goal:

To identify and measure points of "friction" in the customer journey. These views analyze on-site behavior, quantify drop-off points, and track customer dissatisfaction through detailed return metrics.

Query 2.1: Funnel Diagnostics

Description:

This is a diagnostic query used to discover the *true* user journey. It buckets every session into one of 16 possible combinations of actions (e.g., visited_home=0, viewed_product=1, added_to_cart=1, did_purchase=0). This query was critical in proving that our user funnel *does not* start at the homepage, but at the product page.

```sql
-- CX ANALYSIS --------------------------
-- ------ A.On-Site Friction Analysis ---
-- Funnel Drop-off ----------------------

-- identifying the right shcema
WITH SessionFacts AS (
    SELECT
        session_id,
        MAX(CASE WHEN event_type = 'home' THEN 1 ELSE 0 END) AS visited_home,
        MAX(CASE WHEN event_type = 'product' THEN 1 ELSE 0 END) AS viewed_product,
        MAX(CASE WHEN event_type = 'cart' THEN 1 ELSE 0 END) AS added_to_cart,
        MAX(CASE WHEN event_type = 'purchase' THEN 1 ELSE 0 END) AS did_purchase
    FROM
        events
    GROUP BY
        session_id
)

SELECT
    visited_home,
    viewed_product,
    added_to_cart,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

visited_home	viewed_product	added_to_cart	did_purchase	session_count
0	1	1	0	250387
0	1	0	0	249613
0	1	1	1	94047
1	1	1	1	87712

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.2: Funnel Performance & Trend

Description: This query calculates the *real* user funnel (Product -> Cart -> Purchase), which was discovered in the diagnostic view. It calculates the product_to_cart_rate and cart_to_purchase_rate, identifying the two biggest drop-off points. It also trends this performance year-over-year to measure the impact of site improvements.

```
424     -- funnel
425     WITH SessionFacts AS (
426         SELECT
427             session_id,
428             MAX(CASE WHEN event_type = 'product' THEN 1 ELSE 0 END) AS viewed_product,
429             MAX(CASE WHEN event_type = 'cart' THEN 1 ELSE 0 END) AS added_to_cart,
430             MAX(CASE WHEN event_type = 'purchase' THEN 1 ELSE 0 END) AS did_purchase
431         FROM
432             events
433         GROUP BY
434             session_id
435     ),
436     FunnelCounts AS (
437         SELECT
438             SUM(viewed_product) AS step1_viewed_product,
439             SUM(CASE
440                     WHEN viewed_product = 1 AND added_to_cart = 1 THEN 1
441                     ELSE 0
442                 END) AS step2_added_to_cart,
443             SUM(CASE
444                     WHEN viewed_product = 1 AND added_to_cart = 1 AND did_purchase = 1 THEN 1
445                     ELSE 0
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

step1_viewed_product	step2_added_to_cart	product_to_cart_rate	step3_purchased	cart_to_purchase_rate
681759	432146	63.39	181759	42.06

```
474     -- funnel trend
475     WITH SessionFacts AS (
476         SELECT
477             DATE_FORMAT(created_at, '%Y') over_years,
478             session_id,
479             MAX(CASE WHEN event_type = 'product' THEN 1 ELSE 0 END) AS viewed_product,
480             MAX(CASE WHEN event_type = 'cart' THEN 1 ELSE 0 END) AS added_to_cart,
481             MAX(CASE WHEN event_type = 'purchase' THEN 1 ELSE 0 END) AS did_purchase
482         FROM
483             events
484         GROUP BY
485             session_id, DATE_FORMAT(created_at, '%Y')
486     ),
487     FunnelCounts AS (
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

over_years	step1_viewed_product	step2_added_to_cart	product_to_cart_rate	step3_purchased	cart_to_purchase_rate
2019	102094	52938	51.85	3646	6.89
2020	112190	62417	55.64	12668	20.30
2021	123608	74134	59.98	24254	32.72
2022	141246	91921	65.08	42149	45.85
2023	185097	135434	73.17	85064	62.81
2024	17525	15303	87.32	13166	86.04

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.3: Page Exit Rate

Description:

This query creates an "action list" for the CX Manager by identifying the "leakiest" pages on the site. It calculates the exit rate (total_exits / total_views) for every URI, ordered from worst to best. A high exit rate on a page like /cart would indicate a clear friction point driving customers away.

```
460     -- Exit Rate
461     WITH RankedEvents AS (
462         SELECT
463             session_id,
464             uri,
465             ROW_NUMBER() OVER(PARTITION BY session_id ORDER BY sequence_number DESC) AS event_rank
466         FROM
467             events
468     ),
469     ExitCounts AS (
470         SELECT
471             uri,
472             COUNT(*) AS total_exits
473         FROM
474             RankedEvents
475         WHERE
476             event_rank = 1
477             AND uri IS NOT NULL
478         GROUP BY
479             uri
480     ),
481     PageViews AS (
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

uri	total_views	total_exits	exit_rate_percent
cancel	125568	125568	100.00000000
purchase	181759	181759	100.00000000
cart	432146	124819	28.88352540

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.4: State Return Rates

Description:

This view identifies high-risk geographic areas by calculating the order return rate for each state. It uses the safe SUM(CASE... logic to count returns (year(returned_at) <> '0000'). A high rate in a specific state can point to issues with a regional shipping carrier or local fraud.

```
508     -- ---------- B.Dissatisfaction Analysis -------------
509     -- states return rates
510     WITH StateOrders AS (
511         SELECT
512             u.state,
513             o.order_id,
514             o.returned_at
515         FROM
516             orders o
517         JOIN
518             users u ON o.user_id = u.id
519         WHERE
520             u.state IS NOT NULL
521             AND year(o.shipped_at) <> '0000'
522     )
523     SELECT
524         state,
525         COUNT(order_id) AS total_shipped_orders,
526         SUM(CASE
527                 WHEN year(returned_at) <> '0000' THEN 1
528                 ELSE 0
529             END) AS total_returned_orders,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

state	total_shipped_orders	total_returned_orders	return_rate_percent
Dolnośląskie	75	17	22.66666660
Sergipe	171	36	21.05263150
Daejeon	100	21	21.00000000
Brussels	67	14	20.89552230
Schleswig-Holstein	97	20	20.61855670
Gyeongsangbuk-do	64	13	20.31250000
Ningxia Hui Autonomous Region	181	36	19.88950270

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.5: Order Return Rate (ORR) Trend

Description:

This is a CX metric that tracks the % of *orders* that experienced *any* return (even a partial one). A high ORR means many customers are being forced to interact with the returns system, indicating high operational friction. This query tracks the trend yearly.

```
559     -- Order Return Rate - ORR
560     WITH OrderReturnStatus AS (
561         SELECT
562             order_id,
563             MAX(CASE
564                     WHEN status = 'Returned' THEN 1
565                     ELSE 0
566                 END) AS has_a_return
567         FROM
568             order_items
569         GROUP BY
570             order_id
571     )
572     SELECT
573         DATE_FORMAT(o.created_at, '%Y') over_years,
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

over_years	total_shipped_orders	orders_with_a_return	order_return_rate_percent
2019	1674	245	14.63560330
2020	5607	870	15.51631880
2021	10868	1673	15.39381670
2022	19017	2859	15.03391700
2023	38328	5969	15.57347100
2024	5967	914	15.31758000

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.6: Item Return Rate (IRR) Trend

Description:

This is a financial/quality metric that tracks the % of *individual items* that were returned. This is the most accurate measure of the *cost* of returns and is a key performance indicator for the Head of Products and CFO. This query tracks the trend yearly.

```
586     -- Item Return Rate - IRR
587     SELECT
588         DATE_FORMAT(o.created_at, '%Y') over_years,
589         COUNT(oi.id) AS total_shipped_items,
590         SUM(CASE
591                 WHEN oi.status = 'Returned'
592                 THEN 1
593                 ELSE 0
594             END) AS total_returned_items,
595         (CAST(SUM(CASE WHEN oi.status = 'Returned' THEN 1 ELSE 0 END) AS DECIMAL(18,4)) / COUNT(oi.id)) * 100 AS item_return_rate_percent
596     FROM
597         order_items oi
598     JOIN
599         orders o ON oi.order_id = o.order_id
600     WHERE
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

over_years	total_shipped_items	total_returned_items	item_return_rate_percent
2019	2409	355	14.73640510
2020	8236	1309	15.89363760
2021	15798	2419	15.31206480
2022	27551	4106	14.90327020
2023	55615	8707	15.65584820
2024	8672	1336	15.40590400

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.7: Toxic Products

Description:

This query identifies the "Top Toxic Products with 100% return rate" that cause the most customer dissatisfaction. This separates high-volume products with acceptable returns from low-volume products that are truly problematic and damaging to the brand.

```
647     -- Toxic Products
648 *   WITH ItemProductStatus AS (
649         SELECT
650             p.name,
651             oi.id AS item_id,
652             (CASE
653                 WHEN oi.status = 'Returned' THEN 1
654                 ELSE 0
655              END) AS is_returned
656         FROM
657             order_items oi
658         JOIN
659             orders o ON oi.order_id = o.order_id
660         JOIN
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	total_shipped_items	total_returned_items	return_rate_percent
Jones New York Women's Petite 3/4 Sleeve Blouse	2	2	100.00000000
Democracy Women's Triple Collar Woven Surplu...	1	1	100.00000000
Calvin Klein Women's Peasant Blouse	1	1	100.00000000
Allegra K Ladies Button Detail V Neck Close-fittin...	1	1	100.00000000
Foxcroft Women's 3/4 Sleeve Fitted Blouse	1	1	100.00000000
Calvin Klein Jeans Women's Convertible Henley ...	1	1	100.00000000
Gildan G240L Ladies 6.1 oz. Ultra Cotton Long-S...	1	1	100.00000000

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.8: Return Rate by Category

Description:

This query pinpoints systemic product issues by calculating the item return rate (%) for each product *category*. This guides the Head of Products on where to focus quality control efforts.

```
600     -- Return Rate by Category
601     WITH ItemCategoryStatus AS (
602         SELECT
603             p.category,
604             oi.id AS item_id,
605             (CASE
606                 WHEN oi.status = 'Returned' THEN 1
607                 ELSE 0
608              END) AS is_returned
609         FROM
610             order_items oi
611         JOIN
612             orders o ON oi.order_id = o.order_id
613         JOIN
614             products p ON oi.product_id = p.id
615         WHERE
616             year(o.shipped_at) <> '0000'
617     )
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	total_shipped_items	total_returned_items	return_rate_percent
Intimates	8784	1364	15.52823310
Jeans	8234	1237	15.02307500
Fashion Hoodies & Sweatshirts	7766	1217	15.67087300
Tops & Tees	7720	1172	15.18134710
Shorts	7174	1139	15.87677720
Swim	7398	1133	15.31494990
Sweaters	7322	1130	15.43294180
Sleep & Lounge	7206	1060	14.70996390
Active	5979	947	15.83876900
Accessories	6249	945	15.12241950

E-commerce Analysis: Final Scripts & Documentation
CMO / CX

Query 2.9: Return Rate by Distribution Center

Description:

This query identifies operational issues at the warehouse level. It calculates the item return rate (%) for each *Distribution Center*. A high rate from a specific DC (e.g., "New Orleans LA ") while other DCs are normal points to an operational problem (e.g., poor packaging, bad quality control) rather than a product-specific problem.

```
630     -- Return Rate by Geography
631  WITH ShippedItemStatus AS (
632      SELECT
633          oi.id AS item_id,
634          oi.product_id,
635          (CASE
636              WHEN oi.status = 'Returned' THEN 1
637              ELSE 0
638           END) AS is_returned
639      FROM
640          order_items oi
641      JOIN
642          orders o ON oi.order_id = o.order_id
643      WHERE
644          year(o.shipped_at) <> '0000'
645  ),
646
647  ItemDistribution AS (
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

distribution_center_name	total_shipped_items	total_returned_items	return_rate_percent
New Orleans LA	8660	1428	16.48960730
Houston TX	14663	2300	15.68573960
Port Authority of New York/New Jersey NY/NJ	10508	1644	15.64522260
Charleston SC	10951	1682	15.35932790
Los Angeles CA	11262	1729	15.35251280
Memphis TN	15640	2400	15.34526850
Philadelphia PA	11058	1692	15.30113940
Chicago IL	15657	2379	15.19448170
Savannah GA	7786	1181	15.16825070
Mobile AL	12096	1797	14.85615070