```sql
-- Ad Performance Validation Queries (15 Essential Checks)
-- Designed to validate key insights and DAX measures from the Power BI dashboard.
-- Note: Assumes tables are named [ad_events], [ads], [users], [campaigns].

-- ** CRITICAL TIP: Always use CAST(... AS DECIMAL(18, 4)) before division to prevent
-- integer truncation and ensure accurate rate calculations (e.g., CTR, CR, CPA).

-----------------------------------------------------------------------------
-- 1. TOP-LEVEL FUNNEL EFFICIENCY (Validation for overall KPIs and Rates)
-- Validates the main metric cards (Impressions, Clicks, Purchases) and derived rates.
-----------------------------------------------------------------------------
SELECT
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Total_Impressions,
    COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) AS Total_Clicks,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Total_Purchases,

    -- CTR (Click-Through Rate)
    CAST(COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0) AS CTR_Percentage,

    -- Conversion Rate (Purchases / Clicks)
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END), 0) AS ConversionRate_Percentage,

    -- Purchase Rate (Purchases / Impressions)
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0) AS PurchaseRate_Percentage
FROM
    ad_events AS ae;
```

Results | Messages

Total_Impressions	Total_Clicks	Total_Purchases	CTR_Percentage	ConversionRate_Percentage	PurchaseRate_Percentage
339812	40079	2031	11.794462820618459	5.067491703884827	0.597683424952620

```sql
-- ------------------------------------------------------------------------------
-- 2. PERFORMANCE BY PLATFORM (Validation for Facebook vs Instagram effectiveness)
-- Validates the primary split to identify the most effective platform based on PR.
-- ------------------------------------------------------------------------------
SELECT
    a.ad_platform,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Impressions,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Purchases,
    -- Purchase Rate (The most crucial metric for platform comparison)
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0) AS PurchaseRate_Pct
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
GROUP BY a.ad_platform
ORDER BY PurchaseRate_Pct DESC;
```

Results | Messages

ad_platform	Impressions	Purchases	PurchaseRate_Pct
Instagram	74940	454	0.605817987723512
Facebook	162687	974	0.598695654846422

```sql
-- ------------------------------------------------------------------------------
-- 3. SEGMENTED PERFORMANCE BY AD TYPE (Validation for Video/Stories superiority)
-- Validates the insight used for budget reallocation (Video & Stories > Carousel/Image).
-- ------------------------------------------------------------------------------
SELECT
    a.ad_type,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Impressions,
    COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) AS Clicks,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Purchases,
    -- Conversion Rate (CR) Calculation is key here
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END), 0) AS ConversionRate_Percentage
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
GROUP BY a.ad_type
ORDER BY ConversionRate_Percentage DESC;
```

Results | Messages

ad_type	Impressions	Clicks	Purchases	ConversionRate_Percentage
Carousel	54477	6485	347	5.350809560524286
Stories	77783	9156	474	5.176933158584534
Image	62986	7383	374	5.065691453338751
Video	42381	5062	233	4.602923745555116

```sql
-------------------------------------------------------------------------------
-- 4. NEW: ENGAGEMENT RATE BY AD TYPE (Detailed ER Check)
-- Isolates the Engagement Rate to confirm the 'content resonance' insight for each ad type.
-------------------------------------------------------------------------------
SELECT
    a.ad_type,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Impressions,
    COUNT(CASE WHEN ae.event_type IN ('Click', 'Like', 'Share', 'Comment') THEN 1 END) AS Total_Engagements,
    -- Engagement Rate: Total Engagements / Total Impressions
    CAST(COUNT(CASE WHEN ae.event_type IN ('Click', 'Like', 'Share', 'Comment') THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0) AS EngagementRate_Pct
FROM ad_events ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
GROUP BY a.ad_type
ORDER BY EngagementRate_Pct DESC;
```

Results | Messages

ad_type	Impressions	Total_Engagements	EngagementRate_Pct
Video	42381	7294	17.210542460064651
Image	62986	10810	17.162544057409583
Carousel	54477	9340	17.144850120234227
Stories	77783	13308	17.109136957947109

```sql
-----------------------------------------------------------------------------
-- 5. AUDIENCE ENGAGEMENT BY GENDER (Validation for Female vs Male engagement)
-- Validates the gender distribution of total engagements.
-----------------------------------------------------------------------------
SELECT
    u.user_gender,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Impressions,
    COUNT(CASE WHEN ae.event_type IN ('Click', 'Like', 'Share', 'Comment') THEN 1 END) AS Total_Engagements
FROM ad_events AS ae
JOIN users AS u
ON ae.user_id = u.user_id
GROUP BY u.user_gender
ORDER BY Total_Engagements DESC;
```

%

Results | Messages

user_gender	Impressions	Total_Engagements
Male	179326	30637
Female	110790	18967
Other	32627	5670

```sql
-- ------------------------------------------------------------------------------
-- 6. AUDIENCE ENGAGEMENT BY AGE GROUP (Validation for 18-30 being the core audience)
-- Validates which age bracket drives the majority of high-funnel activity (Impressions/Purchases).
-- ------------------------------------------------------------------------------
SELECT
    u.age_group,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Impressions,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Purchases
FROM ad_events AS ae
JOIN users AS u
ON ae.user_id = u.user_id
GROUP BY u.age_group
ORDER BY Impressions DESC;
```

%

Results | Messages

age_group	Impressions	Purchases
25-34	133113	800
18-24	101138	615
35-44	46796	285
16-17	28424	156
45-54	10662	64
55-65	2610	17

```sql
-- ----------------------------------------------------------------------------
-- 7. NEW: USER CONVERSION RATE BY AGE GROUP (Deep Funnel Demographics)
-- Validates the conversion efficiency (Clicks -> Purchase) across age groups to confirm value.
-- ----------------------------------------------------------------------------
SELECT
    u.age_group,
    COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) AS Clicks,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Purchases,
    -- Conversion Rate: Purchases / Clicks
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END), 0) AS ConversionRate_Pct
FROM ad_events AS ae
JOIN users AS u
ON ae.user_id = u.user_id
GROUP BY u.age_group
ORDER BY ConversionRate_Pct DESC;
```

Results | Messages

age_group	Clicks	Purchases	ConversionRate_Pct
55-65	297	17	5.723905723905723
35-44	5495	285	5.186533212010919
18-24	11877	615	5.178075271533215
25-34	15789	800	5.066818671226803
45-54	1289	64	4.965089216446858
16-17	3344	156	4.665071770334928

```sql
-- ------------------------------------------------------------------------------
-- 8. GEOGRAPHIC SEGMENTATION (Volume vs. Value Countries)
-- Validates the split between high-volume (Impressions) and high-value (Purchases/PR) regions.
-- ------------------------------------------------------------------------------
SELECT
    u.country,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Total_Impressions,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Total_Purchases,
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0) AS PurchaseRate_Pct
FROM ad_events AS ae
JOIN users AS u
ON ae.user_id = u.user_id
GROUP BY u.country
HAVING -- Filter for countries with at least a meaningful number of impressions
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) > 500
ORDER BY Total_Purchases DESC;
```

%

Results | Messages

country	Total_Impressions	Total_Purchases	PurchaseRate_Pct
United States	97336	635	0.652379386866113
United Kingdom	48965	271	0.553456550597365
Canada	32139	192	0.597405021935965
India	30268	176	0.581472181842209
Germany	26980	146	0.541141586360266
Australia	22944	118	0.514295676429567
Japan	15945	117	0.733772342427093
Brazil	19425	113	0.581724581724581
Mexico	16435	113	0.687557042896257
France	12306	56	0.455062571103526

```sql
-- ----------------------------------------------------------------------------
-- 9. TIME-BASED VALIDATION (Daily Trend - Day of Week)
-- Validates the daily performance pattern (e.g., higher activity on weekends or weekdays).
-- ----------------------------------------------------------------------------
SELECT
    ae.day_of_week, -- Field assumed to exist in ad_events
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Total_Impressions,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Total_Purchases
FROM ad_events AS ae
GROUP BY
    ae.day_of_week
-- Ensure days are ordered correctly (e.g., using a CASE statement or DATEPART if needed)
ORDER BY
    CASE ae.day_of_week
        WHEN 'Sunday' THEN 1 WHEN 'Monday' THEN 2 WHEN 'Tuesday' THEN 3 WHEN 'Wednesday' THEN 4
        WHEN 'Thursday' THEN 5 WHEN 'Friday' THEN 6 WHEN 'Saturday' THEN 7 END;
```

Results | Messages

day_of_week	Total_Impressions	Total_Purchases
Sunday	48426	281
Monday	48597	292
Tuesday	48474	268
Wednesday	48498	273
Thursday	48598	314
Friday	48759	308
Saturday	48460	295

```sql
-------------------------------------------------------------------------------
-- 10. COST PER METRIC (Financial Validation for CPC & CPM)
-- Validates the cost efficiency measures. Requires joining to the [campaigns] table.
-------------------------------------------------------------------------------
SELECT
    SUM(c.total_budget) AS Total_Ad_Budget,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Total_Impressions,
    COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) AS Total_Clicks,

    -- CPC (Cost Per Click): Total Budget / Total Clicks
    SUM(c.total_budget) / NULLIF(COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END), 0) AS Avg_CPC,

    -- CPM (Cost Per Mille/1000 Impressions): (Total Budget / Total Impressions) * 1000
    (SUM(c.total_budget) / NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0)) * 1000 AS Avg_CPM
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
JOIN campaigns AS c
ON a.campaign_id = c.campaign_id;
```

%

Results | Messages

Total_Ad_Budget	Total_Impressions	Total_Clicks	Avg_CPC	Avg_CPM
14316476630.1465	237627	28086	509737.115650021	60247684.9438258

```sql
-------------------------------------------------------------------------------
-- 11. NEW: COST PER ACQUISITION (CPA) BY PLATFORM (Deep Financial Metric)
-- Validates the true cost of acquiring a purchasing customer on Facebook vs. Instagram.
-------------------------------------------------------------------------------
SELECT
    a.ad_platform,
    SUM(c.total_budget) AS Platform_Budget,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Total_Purchases,
    -- CPA (Cost Per Acquisition): Platform Budget / Total Purchases
    SUM(c.total_budget) / NULLIF(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END), 0) AS Avg_CPA
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
JOIN campaigns AS c
ON a.campaign_id = c.campaign_id
GROUP BY a.ad_platform
ORDER BY Avg_CPA ASC; -- Lower CPA is better
```

%

Results | Messages

ad_platform	Platform_Budget	Total_Purchases	Avg_CPA
Facebook	9704185595.36621	974	9963229.56403102
Instagram	4612291034.78027	454	10159231.3541416

```sql
-----------------------------------------------------------------------------
-- 12. CAMPAIGN PERFORMANCE RANKING (Identifying highest Purchase Rate campaigns)
-- Validates the list/ranking visualization used to determine most efficient campaigns.
-----------------------------------------------------------------------------
SELECT TOP 10
    c.name AS Campaign_Name,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Total_Impressions,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Total_Purchases,

    -- Purchase Rate (Best measure of campaign success)
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END), 0) AS PurchaseRate_Pct
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
JOIN campaigns AS c
ON a.campaign_id = c.campaign_id
GROUP BY c.name
HAVING -- Only include campaigns with significant activity (e.g., at least 500 impressions)
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) >= 500
ORDER BY PurchaseRate_Pct DESC;
```

%

Results | Messages

ad_platform	Platform_Budget	Total_Purchases	Avg_CPA
Facebook	9704185595.36621	974	9963229.56403102
Instagram	4612291034.78027	454	10159231.3541416

```sql
-- ------------------------------------------------------------------------------
-- 13. NEW: CAMPAIGN DURATION VS. PURCHASE RATE
-- Validates if shorter, focused campaigns or longer, sustained campaigns yield better results.
-- ------------------------------------------------------------------------------
SELECT
    CASE
        WHEN c.duration_days <= 30 THEN 'Short (<=30 Days)'
        WHEN c.duration_days BETWEEN 31 AND 60 THEN 'Medium (31-60 Days)'
        ELSE 'Long (>60 Days)'
    END AS Campaign_Duration_Group,
    COUNT(DISTINCT c.campaign_id) AS Number_of_Campaigns,
    SUM(CASE WHEN ae.event_type = 'Impression' THEN 1 ELSE 0 END) AS Total_Impressions,
    SUM(CASE WHEN ae.event_type = 'Purchase' THEN 1 ELSE 0 END) AS Total_Purchases,

    CAST(SUM(CASE WHEN ae.event_type = 'Purchase' THEN 1 ELSE 0 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(SUM(CASE WHEN ae.event_type = 'Impression' THEN 1 ELSE 0 END), 0) AS PurchaseRate_Pct
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
JOIN campaigns AS c
ON a.campaign_id = c.campaign_id
GROUP BY
    CASE
        WHEN c.duration_days <= 30 THEN 'Short (<=30 Days)'
        WHEN c.duration_days BETWEEN 31 AND 60 THEN 'Medium (31-60 Days)'
        ELSE 'Long (>60 Days)'
    END
ORDER BY PurchaseRate_Pct DESC;
```

%

Results | Messages

Campaign_Duration_Group	Number_of_Campaigns	Total_Impressions	Total_Purchases	PurchaseRate_Pct
Long (>60 Days)	29	145709	887	0.608747572215854
Medium (31-60 Days)	18	91918	541	0.588568071542026

```sql
-- ----------------------------------------------------------------------------
-- 14. NEW: PERFORMANCE BY TARGET INTEREST (Top Performing Interests)
-- Validates the primary interest categories that drive the best Conversion Rate.
-- ----------------------------------------------------------------------------
-- Note: This assumes interests in the 'ads' table are stored as single values or a primary interest can be extracted.
-- If 'target_interests' is a comma-separated string, a more complex string parsing/unnesting function would be required.
SELECT
    a.target_interests,
    COUNT(CASE WHEN ae.event_type = 'Impression' THEN 1 END) AS Impressions,
    COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) AS Clicks,
    COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS Purchases,
    -- Conversion Rate: Purchases / Clicks
    CAST(COUNT(CASE WHEN ae.event_type = 'Purchase' THEN 1 END) AS DECIMAL(18, 4)) * 100 /
    NULLIF(COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END), 0) AS ConversionRate_Pct
FROM ad_events AS ae
JOIN ads_1 AS a
ON ae.ad_id = a.ad_id
GROUP BY a.target_interests
HAVING
    COUNT(CASE WHEN ae.event_type = 'Click' THEN 1 END) >= 100 -- Filter for statistically significant data
ORDER BY ConversionRate_Pct DESC;
```

%

Results | Messages

target_interests	Impressions	Clicks	Purchases	ConversionRate_Pct
fitness, art	1717	210	18	8.571428571428571
news, travel	1698	195	15	7.692307692307692
fitness, sports	1700	199	15	7.537688442211055
health, finance	1748	188	14	7.446808510638297
sports, photo...	1718	207	15	7.246376811594202
lifestyle, news	1756	195	14	7.179487179487179
art, travel	1698	206	14	6.796116504854368
fashion, gaming	5198	583	39	6.689536878216123
gaming, travel	3407	365	24	6.575342465753424
gaming, tech...	1688	214	14	6.542056074766355
travel, finance	1729	185	12	6.486486486486486
news, fashion	1709	216	14	6.481481481481481
fitness, gaming	1643	187	12	6.417112299465240
travel, photog...	1751	205	13	6.341463414634146

```sql
-- ------------------------------------------------------------------------------
-- 15. VALIDATION OF RETARGETING AUDIENCE (Users who clicked but did not purchase)
-- Identifies the size of the key audience segment for retargeting efforts.
-- ------------------------------------------------------------------------------
WITH Clicked_Users AS (
    SELECT DISTINCT user_id FROM ad_events WHERE event_type = 'Click'
),
Purchased_Users AS (
    SELECT DISTINCT user_id FROM ad_events WHERE event_type = 'Purchase'
)
SELECT
    COUNT(c.user_id) AS Users_Clicked_Not_Purchased -- This is the size of the retargeting pool
FROM Clicked_Users AS c
LEFT JOIN Purchased_Users AS p
ON c.user_id = p.user_id
WHERE p.user_id IS NULL;
```

%

Results | Messages

Users_Clicked_Not_Purchased
7966