

Project Title: E-commerce & Web Analytics: **Maximizing Fuzzy Factory's Digital Potential - Advanced** **SQL**

Background:

Fuzzy Factory, an online retailer, seeks to optimize its marketing channels and website performance following the launch of a new product. With a vast database of over 2 million records, the project focuses on leveraging MySQL to analyze and optimize marketing strategies, website conversion rates, and product portfolio performance.

Project Objectives:

The Maven Fuzzy Factory Marketing and Website Optimization Analysis project aims to:

1. Analyze and optimize marketing channels, including Gsearch campaigns, to maximize business growth potential.
2. Measure and test website conversion performance to enhance user experience and drive higher conversion rates.
3. Understand the impact of new product launches on overall business performance and customer engagement.
4. Utilize MySQL to access and explore the Maven Fuzzy Factory database, enabling comprehensive analysis of marketing channels, website performance, and product portfolio.

Project Queries:

1. Analyze monthly trends for Gsearch sessions and orders to showcase growth.

Query:

```
SELECT
-- YEAR(created_at),
-- MONTH(created_at),
MIN(DATE(ws.created_at)) AS month_start_date,
COUNT(DISTINCT(ws.website_session_id)) AS total_sessions,
COUNT(DISTINCT(o.order_id)) AS total_orders,
COUNT(DISTINCT(o.order_id))/ COUNT(DISTINCT(ws.website_session_id)) AS conv_rate
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2012-11-27' AND ws.utm_source = 'gsearch'
GROUP BY
YEAR(ws.created_at),
MONTH(ws.created_at);
```

Output:

	month_start_date	total_sessions	total_orders	conv_rate
▶	2012-03-19	1860	60	0.0323
	2012-04-01	3574	92	0.0257
	2012-05-01	3410	97	0.0284
	2012-06-01	3578	121	0.0338
	2012-07-01	3811	145	0.0380
	2012-08-01	4877	184	0.0377
	2012-09-01	4491	188	0.0419
	2012-10-01	5534	234	0.0423
	2012-11-01	8889	373	0.0420

- Assess the performance of Gsearch campaigns, specifically nonbrand and brand campaigns, by monthly trends.

Query:

```
SELECT
  -- YEAR(created_at),
  -- MONTH(created_at),
  MIN(DATE(ws.created_at)) AS month_start_date,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN ws.website_session_id ELSE NULL END) AS nonbrand_sessions,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END) AS nonbrand_orders,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN ws.website_session_id ELSE NULL END) AS brand_sessions,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN o.order_id ELSE NULL END) AS brand_orders
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2012-11-27' AND ws.utm_source = 'gsearch'
GROUP BY
  YEAR(ws.created_at),
  MONTH(ws.created_at);
```

Output:

	month_start_date	nonbrand_sessions	nonbrand_orders	brand_sessions	brand_orders
►	2012-03-19	1852	60	8	0
	2012-04-01	3509	86	65	6
	2012-05-01	3295	91	115	6
	2012-06-01	3439	114	139	7
	2012-07-01	3660	136	151	9
	2012-08-01	4673	174	204	10
	2012-09-01	4227	172	264	16
	2012-10-01	5197	219	337	15
	2012-11-01	8506	356	383	17

3. Dive into nonbrand Gsearch sessions and orders, segmented by device type, to understand traffic sources.

Query:

```
SELECT
  -- YEAR(created_at),
  -- MONTH(created_at),
  MIN(DATE(ws.created_at)) AS month_start_date,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND device_type = 'desktop' THEN ws.website_session_id ELSE NULL END) AS nonbrand_desktop_sessions,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND device_type = 'desktop' THEN o.order_id ELSE NULL END) AS nonbrand_desktop_orders,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND device_type = 'mobile' THEN ws.website_session_id ELSE NULL END) AS nonbrand_mobile_sessions,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND device_type = 'mobile' THEN o.order_id ELSE NULL END) AS nonbrand_mobile_orders,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' AND device_type = 'desktop' THEN ws.website_session_id ELSE NULL END) AS brand_desktop_sessions,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' AND device_type = 'desktop' THEN o.order_id ELSE NULL END) AS brand_desktop_orders,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' AND device_type = 'mobile' THEN ws.website_session_id ELSE NULL END) AS brand_mobile_sessions,
  COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' AND device_type = 'mobile' THEN o.order_id ELSE NULL END) AS brand_mobile_orders
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2012-11-27' AND ws.utm_source = 'gsearch'
GROUP BY
  YEAR(ws.created_at),
  MONTH(ws.created_at);
```

Output:

	month_start_date	nonbrand_desktop_sessions	nonbrand_desktop_orders	nonbrand_mobile_sessions	nonbrand_mobile_orders	brand_desktop_sessions	brand_desktop_orders	brand_mobile_sessions	brand_mobile_orders
▶	2012-03-19	1128	50	724	10	5	0	3	0
	2012-04-01	2139	75	1370	11	42	5	23	1
	2012-05-01	2276	83	1019	8	72	4	43	2
	2012-06-01	2673	106	766	8	88	7	51	0
	2012-07-01	2774	122	886	14	82	8	69	1
	2012-08-01	3515	165	1158	9	124	9	80	1
	2012-09-01	3171	155	1056	17	154	14	110	2
	2012-10-01	3934	201	1263	18	214	12	123	3
	2012-11-01	6457	323	2049	33	221	13	162	4

4. Compare monthly trends for Gsearch with other marketing channels to address concerns about traffic dependency.

Query:

```
SELECT
    YEAR(created_at) AS yr,
    MONTH(created_at) AS mon,
    COUNT(website_session_id) AS total_sessions,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' THEN website_session_id ELSE NULL END) AS total_gsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' THEN website_session_id ELSE NULL END) AS total_bsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source = 'socialbook' THEN website_session_id ELSE NULL END) AS total_socialbook_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN website_session_id ELSE NULL END) AS total_organic_search_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN website_session_id ELSE NULL END) AS total_direct_typein_sessions
FROM website_sessions
WHERE created_at < '2012-11-27'
GROUP BY 1,2;
```

Output:

	yr	mon	total_sessions	total_gsearch_paid_sessions	total_bsearch_paid_sessions	total_socialbook_paid_sessions	total_organic_search_sessions	total_direct_typein_sessions
▶	2012	3	1879	1860	2	0	8	9
	2012	4	3734	3574	11	0	78	71
	2012	5	3736	3410	25	0	150	151
	2012	6	3963	3578	25	0	190	170
	2012	7	4249	3811	44	0	207	187
	2012	8	6097	4877	705	0	265	250
	2012	9	6546	4491	1439	0	331	285
	2012	10	8183	5534	1781	0	428	440
	2012	11	12750	8889	2840	0	536	485

5. Evaluate website performance improvements over eight months by analyzing session-to-order conversion rates.

Query:

```
SELECT
    YEAR(ws.created_at) AS yrs,
    MONTH(ws.created_at) AS mon,
    COUNT(DISTINCT ws.website_session_id) AS total_sessions,
    COUNT(DISTINCT o.order_id) AS total_orders,
    COUNT(DISTINCT o.order_id)/COUNT(DISTINCT ws.website_session_id) AS conv_rate
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2012-11-27'
GROUP BY
    YEAR(created_at),
    MONTH(created_at);
```

Output:

	yrs	mon	total_sessions	total_orders	conv_rate
►	2012	3	1879	60	0.0319
	2012	4	3734	99	0.0265
	2012	5	3736	108	0.0289
	2012	6	3963	140	0.0353
	2012	7	4249	169	0.0398
	2012	8	6097	228	0.0374
	2012	9	6546	287	0.0438
	2012	10	8183	371	0.0453
	2012	11	12750	561	0.0440

6. Estimate revenue generated from a Gsearch lander test and assess its incremental value.

Query:

```
WITH lander_test_filter AS(
SELECT
    wp.website_session_id,
    wp.created_at,
    wp.pageview_url,
    o.order_id,
    o.price_usd,
    o.cogs_usd
FROM website_pageviews AS wp
LEFT JOIN orders AS o
ON wp.website_session_id = o.website_session_id
LEFT JOIN website_sessions AS ws
ON ws.website_session_id = wp.website_session_id
WHERE wp.created_at > '2012-06-19'
AND wp.created_at < '2012-07-28'
AND wp.pageview_url IN ('/home', '/lander-1')
AND ws.utm_source = 'gsearch'
AND ws.utm_campaign = 'nonbrand'
)
SELECT
    pageview_url,
    COUNT(DISTINCT website_session_id) AS total_sessions,
    COUNT(DISTINCT order_id) AS total_order,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id) AS conversion_rate
FROM lander_test_filter
GROUP BY 1;
```

Output:

	pageview_url	total_sessions	total_order	conversion_rate
▶	/home	2261	72	0.0318
	/lander-1	2316	94	0.0406

7. Showcase a full conversion funnel from landing pages to orders for a previous landing page test.

Query:

```
WITH all_filtered_data AS(
SELECT
    ws.website_session_id,
    wp.pageview_url,
    CASE WHEN wp.pageview_url = '/home' THEN 1 ELSE 0 END AS home_page,
    CASE WHEN wp.pageview_url = '/lander-1' THEN 1 ELSE 0 END AS lander_page,
    CASE WHEN wp.pageview_url = '/products' THEN 1 ELSE 0 END AS products_page,
    CASE WHEN wp.pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS mrfuzzy_page,
    CASE WHEN wp.pageview_url = '/cart' THEN 1 ELSE 0 END AS cart_page,
    CASE WHEN wp.pageview_url = '/shipping' THEN 1 ELSE 0 END AS shipping_page,
    CASE WHEN wp.pageview_url = '/billing' THEN 1 ELSE 0 END AS billing_page,
    CASE WHEN wp.pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS thankyou_page
FROM website_pageviews AS wp
LEFT JOIN website_sessions AS ws
ON wp.website_session_id = ws.website_session_id
WHERE wp.created_at > '2012-06-19'
AND wp.created_at < '2012-07-28'
AND ws.utm_source = 'gsearch'
AND ws.utm_campaign = 'nonbrand'),

max_data AS(
SELECT
    website_session_id,
    MAX(home_page) AS max_home_page,
    MAX(lander_page) AS max_lander_page,
    MAX(products_page) AS max_product_page,
    MAX(mrfuzzy_page) AS max_mrfuzzy_page,
    MAX(cart_page) AS max_cart_page,
    MAX(shipping_page) AS max_shipping_page,
    MAX(billing_page) AS max_billing_page,
    MAX(thankyou_page) AS max_thankyou_page
FROM all_filtered_data
GROUP BY 1),

max_home_filtered_data AS (
SELECT *
FROM max_data
WHERE max_home_page = 1),
max_lander_filtered_data AS (
SELECT *
FROM max_data
WHERE max_lander_page = 1),
```



```

> home_clicks AS(
SELECT
CASE WHEN max_home_page = 1 THEN 'home_page' ELSE NULL END AS segment,
COUNT(CASE WHEN max_home_page = 1 THEN website_session_id ELSE NULL END) AS click_home_page,
COUNT(CASE WHEN max_product_page = 1 THEN website_session_id ELSE NULL END) AS click_product_page,
COUNT(CASE WHEN max_mrfuzzy_page = 1 THEN website_session_id ELSE NULL END) AS click_mrfuzzy_page,
COUNT(CASE WHEN max_cart_page = 1 THEN website_session_id ELSE NULL END) AS click_cart_page,
COUNT(CASE WHEN max_shipping_page = 1 THEN website_session_id ELSE NULL END) AS click_shipping_page,
COUNT(CASE WHEN max_billing_page = 1 THEN website_session_id ELSE NULL END) AS click_billing_page,
COUNT(CASE WHEN max_thankyou_page = 1 THEN website_session_id ELSE NULL END) AS click_thankyou_page
FROM max_home_filtered_data),

> lander_clicks AS(
SELECT
CASE WHEN max_lander_page =1 THEN 'lander_page' ELSE NULL END AS segment,
COUNT(CASE WHEN max_lander_page = 1 THEN website_session_id ELSE NULL END) AS click_home_page,
COUNT(CASE WHEN max_product_page = 1 THEN website_session_id ELSE NULL END) AS click_product_page,
COUNT(CASE WHEN max_mrfuzzy_page = 1 THEN website_session_id ELSE NULL END) AS click_mrfuzzy_page,
COUNT(CASE WHEN max_cart_page = 1 THEN website_session_id ELSE NULL END) AS click_cart_page,
COUNT(CASE WHEN max_shipping_page = 1 THEN website_session_id ELSE NULL END) AS click_shipping_page,
COUNT(CASE WHEN max_billing_page = 1 THEN website_session_id ELSE NULL END) AS click_billing_page,
COUNT(CASE WHEN max_thankyou_page = 1 THEN website_session_id ELSE NULL END) AS click_thankyou_page
FROM max_lander_filtered_data),

click_funnels AS(
SELECT *
FROM home_clicks
UNION ALL
SELECT *
FROM lander_clicks)

SELECT
segment,
click_home_page AS home_clicks,
click_product_page/click_home_page AS product_click_rate,
click_mrfuzzy_page/click_product_page AS mrfuzzy_click_rate,
click_cart_page/click_mrfuzzy_page AS cart_click_rate,
click_shipping_page/click_cart_page AS shipping_click_rate,
click_billing_page/click_shipping_page AS billing_click_rate,
click_thankyou_page/click_billing_page AS thankyou_click_rate
FROM click_funnels
;

```

Output:

	segment	home_clicks	product_click_rate	mrfuzzy_click_rate	cart_click_rate	shipping_click_rate	billing_click_rate	thankyou_click_rate
▶	home_page	2261	0.4166	0.7261	0.4327	0.6757	0.8400	0.4286
	lander_page	2316	0.4676	0.7128	0.4508	0.6638	0.8528	0.4772

8. Quantify the impact of a billing test in terms of revenue per billing page session and monthly impact.

Query:

```
WITH filtered_data AS (  
  SELECT  
    wp.website_session_id,  
    wp.pageview_url,  
    o.price_usd  
  FROM website_pageviews AS wp  
  LEFT JOIN orders AS o  
  ON wp.website_session_id = o.website_session_id  
  WHERE wp.created_at > '2012-09-10'  
  AND wp.created_at < '2012-11-10'  
  AND wp.pageview_url IN ('/billing', '/billing-2'))  
  
avg_prices AS (  
  SELECT  
    pageview_url,  
    COUNT( DISTINCT website_session_id) AS sessions,  
    SUM(price_usd)/ COUNT( DISTINCT website_session_id) AS avg_price_per_billing_pg  
  FROM filtered_data  
  GROUP BY pageview_url)  
  
-- Average price with the billing-2 increased from 22.8 for billing to 31.3 for billing-2 a difference of 8.5 was seen  
-- Total Revenue after the billing test we need to calculate the total sessions after the test  
  
SELECT  
  COUNT( DISTINCT website_session_id) AS total_sessions  
FROM website_pageviews  
WHERE created_at > '2012-10-27'  
AND created_at < '2012-11-27'  
AND pageview_url IN ('/billing', '/billing-2');  
  
-- Total revenue for the last month will be 8.5 * total sessions i.e 1193
```

Output:

	total_sessions
▶	1193

9. Track overall session and order volume trended by quarter to demonstrate volume growth.

Query:

```
SELECT
    YEAR(ws.created_at) AS yr,
    QUARTER(ws.created_at) AS qtr,
    COUNT(ws.website_session_id) AS total_sessions,
    COUNT(o.order_id) AS total_orders
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2015-03-20'
GROUP BY 1,2;
```

Output:

	yr	qtr	total_sessions	total_orders
►	2012	1	1879	60
	2012	2	11433	347
	2012	3	16892	684
	2012	4	32266	1495
	2013	1	19833	1273
	2013	2	24745	1718
	2013	3	27663	1840
	2013	4	40540	2616
	2014	1	46779	3069
	2014	2	53129	3848
	2014	3	57141	4035
	2014	4	76373	5908
	2015	1	64198	5420

10. Present quarterly figures for session-to-order conversion rate, revenue per order, and revenue per session to highlight efficiency improvements.

Query:

```
SELECT
    YEAR(ws.created_at) AS yr,
    QUARTER(ws.created_at) AS qtr,
    COUNT(ws.website_session_id) AS total_sessions,
    COUNT(o.order_id) AS total_orders,
    COUNT(o.order_id)/COUNT(ws.website_session_id) AS conv_rate,
    SUM(o.price_usd) AS total_revenue,
    SUM(o.price_usd)/COUNT(o.order_id) AS revenue_per_order,
    SUM(o.price_usd)/COUNT(ws.website_session_id) AS revenue_per_session
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2015-03-20'
GROUP BY 1,2;
```

Output:

	yr	qtr	total_sessions	total_orders	conv_rate	total_revenue	revenue_per_order	revenue_per_session
►	2012	1	1879	60	0.0319	2999.40	49.990000	1.596275
	2012	2	11433	347	0.0304	17346.53	49.990000	1.517233
	2012	3	16892	684	0.0405	34193.16	49.990000	2.024222
	2012	4	32266	1495	0.0463	74735.05	49.990000	2.316217
	2013	1	19833	1273	0.0642	66377.27	52.142396	3.346809
	2013	2	24745	1718	0.0694	88542.82	51.538312	3.578211
	2013	3	27663	1840	0.0665	95191.54	51.734533	3.441114
	2013	4	40540	2616	0.0645	143136.24	54.715688	3.530741
	2014	1	46779	3069	0.0656	190771.14	62.160684	4.078136
	2014	2	53129	3848	0.0724	247711.95	64.374207	4.662462
	2014	3	57141	4035	0.0706	260237.12	64.494949	4.554298
	2014	4	76373	5908	0.0774	376891.98	63.793497	4.934885
	2015	1	64198	5420	0.0844	340375.55	62.799917	5.301965

11. Analyze quarterly orders from specific channels to demonstrate channel growth.

Query:

```
SELECT
  YEAR(ws.created_at) AS yr,
  QUARTER(ws.created_at) AS qtr,
  COUNT( CASE WHEN ws.utm_source = 'gsearch' and utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END) AS gsearch_nonbrand_orders,
  COUNT( CASE WHEN ws.utm_source = 'bsearch' and utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END) AS bsearch_nonbrand_orders,
  COUNT( CASE WHEN ws.utm_campaign = 'brand' THEN o.order_id ELSE NULL END) AS brand_orders,
  COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NOT NULL THEN o.order_id ELSE NULL END) AS organic_search_orders,
  COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NULL THEN o.order_id ELSE NULL END) AS direct_typein_orders
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2015-03-20'
GROUP BY 1,2;
```

Output:

	yr	qtr	gsearch_nonbrand_orders	bsearch_nonbrand_orders	brand_orders	organic_search_orders	direct_typein_orders
►	2012	1	60	0	0	0	0
	2012	2	291	0	20	15	21
	2012	3	482	82	48	40	32
	2012	4	913	311	88	94	89
	2013	1	766	183	108	125	91
	2013	2	1114	237	114	134	119
	2013	3	1132	245	153	167	143
	2013	4	1657	291	248	223	197
	2014	1	1667	344	354	338	311
	2014	2	2208	427	410	436	367
	2014	3	2259	434	432	445	402
	2014	4	3248	683	615	605	532
	2015	1	3025	581	622	640	552

12. Evaluate session-to-order conversion rate trends for channels by quarter, noting major improvements.

Query:

```
SELECT
    YEAR(ws.created_at) AS yr,
    QUARTER(ws.created_at) AS qtr,
    COUNT( CASE WHEN ws.utm_source = 'gsearch' and utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END) AS gsearch_nonbrand_orders,
    COUNT( CASE WHEN ws.utm_source = 'gsearch' and utm_campaign = 'nonbrand' THEN ws.website_session_id ELSE NULL END) AS gsearch_nonbrand_sessions,
    COUNT( CASE WHEN ws.utm_source = 'gsearch' and utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END)/
    COUNT( CASE WHEN ws.utm_source = 'gsearch' and utm_campaign = 'nonbrand' THEN ws.website_session_id ELSE NULL END) AS gsearch_nonbrand_conv_rate,
    COUNT( CASE WHEN ws.utm_source = 'bsearch' and utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END) AS bsearch_nonbrand_orders,
    COUNT( CASE WHEN ws.utm_source = 'bsearch' and utm_campaign = 'nonbrand' THEN ws.website_session_id ELSE NULL END) AS bsearch_nonbrand_sessions,
    COUNT( CASE WHEN ws.utm_source = 'bsearch' and utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END)/
    COUNT( CASE WHEN ws.utm_source = 'bsearch' and utm_campaign = 'nonbrand' THEN ws.website_session_id ELSE NULL END) AS bsearch_nonbrand_conv_rate,
    COUNT( CASE WHEN ws.utm_campaign = 'brand' THEN o.order_id ELSE NULL END) AS brand_orders,
    COUNT( CASE WHEN ws.utm_campaign = 'brand' THEN ws.website_session_id ELSE NULL END) AS brand_sessions,
    COUNT( CASE WHEN ws.utm_campaign = 'brand' THEN o.order_id ELSE NULL END)/COUNT( CASE WHEN ws.utm_campaign = 'brand' THEN ws.website_session_id ELSE NULL END) AS brand_session_conv_rate,
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NOT NULL THEN o.order_id ELSE NULL END) AS organic_search_orders,
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NOT NULL THEN ws.website_session_id ELSE NULL END) AS organic_search_sessions,
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NOT NULL THEN o.order_id ELSE NULL END)/
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NOT NULL THEN ws.website_session_id ELSE NULL END) AS organic_search_conv_rate,
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NULL THEN o.order_id ELSE NULL END) AS direct_typein_orders,
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NULL THEN ws.website_session_id ELSE NULL END) AS direct_typein_sessions,
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NULL THEN o.order_id ELSE NULL END)/
    COUNT( CASE WHEN ws.utm_source IS NULL AND ws.http_referer IS NULL THEN ws.website_session_id ELSE NULL END) AS direct_typein_conv_rate
FROM website_sessions AS ws
LEFT JOIN orders AS o
ON ws.website_session_id = o.website_session_id
WHERE ws.created_at < '2015-03-20'
GROUP BY 1,2;
```

Output:

yr	qtr	gsearch_nonbrand_orders	gsearch_nonbrand_sessions	gsearch_nonbrand_conv_rate	bsearch_nonbrand_orders	bsearch_nonbrand_sessions	bsearch_nonbrand_conv_rate	brand_orders	brand_sessions	brand_session_conv_rate	organic_search_orders	organic_search_sessions	organic_search_conv_rate
2012	1	60	1852	0.0324	0	0	NULL	0	10	0.0000	0	8	0.0000
2012	2	291	10243	0.0284	0	0	NULL	20	380	0.0526	15	418	0.0359
2012	3	482	12560	0.0384	82	2009	0.0408	48	798	0.0602	40	803	0.0498
2012	4	913	20949	0.0436	311	6260	0.0497	88	1656	0.0531	94	1744	0.0539
2013	1	766	12512	0.0612	183	2642	0.0693	108	1536	0.0703	125	1661	0.0753
2013	2	1114	16251	0.0685	237	3433	0.0690	114	1680	0.0679	134	1763	0.0760
2013	3	1132	17705	0.0639	245	3516	0.0697	153	2176	0.0703	167	2276	0.0734
2013	4	1657	26338	0.0629	291	4845	0.0601	248	3097	0.0801	223	3214	0.0694
2014	1	1667	24045	0.0693	344	4887	0.0704	354	4217	0.0839	338	4471	0.0756
2014	2	2208	31442	0.0702	427	6144	0.0695	410	5102	0.0804	436	5468	0.0797
2014	3	2259	32142	0.0703	434	6217	0.0698	432	5714	0.0756	445	6075	0.0733
2014	4	3248	41525	0.0782	683	8119	0.0841	615	7574	0.0812	605	7718	0.0784
2015	1	3025	35142	0.0861	581	6837	0.0850	622	7303	0.0852	640	7792	0.0821

nonbrand_conv_rate	bsearch_nonbrand_orders	bsearch_nonbrand_sessions	bsearch_nonbrand_conv_rate	brand_orders	brand_sessions	brand_session_conv_rate	organic_search_orders	organic_search_sessions	organic_search_conv_rate	direct_typein_orders	direct_typein_sessions	direct_typein_conv_rate
0	0	0	0.0000	0	10	0.0000	0	8	0.0000	0	9	0.0000
0	0	0	0.0000	20	380	0.0526	15	418	0.0359	21	392	0.0536
82	2009	0.0408	0.0408	48	798	0.0602	40	803	0.0498	32	722	0.0443
311	6260	0.0497	0.0497	88	1656	0.0531	94	1744	0.0539	89	1657	0.0537
183	2642	0.0693	0.0693	108	1536	0.0703	125	1661	0.0753	91	1482	0.0614
237	3433	0.0690	0.0690	114	1680	0.0679	134	1763	0.0760	119	1618	0.0735
245	3516	0.0697	0.0697	153	2176	0.0703	167	2276	0.0734	143	1990	0.0719
291	4845	0.0601	0.0601	248	3097	0.0801	223	3214	0.0694	197	3046	0.0647
344	4887	0.0704	0.0704	354	4217	0.0839	338	4471	0.0756	311	4064	0.0765
427	6144	0.0695	0.0695	410	5102	0.0804	436	5468	0.0797	367	4973	0.0738
434	6217	0.0698	0.0698	432	5714	0.0756	445	6075	0.0733	402	5726	0.0702
683	8119	0.0841	0.0841	615	7574	0.0812	605	7718	0.0784	532	7114	0.0748
581	6837	0.0850	0.0850	622	7303	0.0852	640	7792	0.0821	552	7124	0.0775

13. Analyze monthly revenue and margin by product, along with total sales and revenue, noting seasonality trends.

Query:

```
SELECT *
FROM ORDERS;

SELECT
    YEAR(created_at) AS yrs,
    MONTH(created_at) AS mnth,
    SUM(price_usd) AS total_revenue,
    SUM(price_usd)-SUM(cogs_usd) AS total_margin,
    SUM( CASE WHEN primary_product_id = 1 THEN price_usd ELSE NULL END) AS product_1_revenue,
    SUM( CASE WHEN primary_product_id = 1 THEN price_usd - cogs_usd ELSE NULL END) AS product_1_margin,
    SUM( CASE WHEN primary_product_id = 2 THEN price_usd ELSE NULL END) AS product_2_revenue,
    SUM( CASE WHEN primary_product_id = 2 THEN price_usd - cogs_usd ELSE NULL END) AS product_2_margin,
    SUM( CASE WHEN primary_product_id = 3 THEN price_usd ELSE NULL END) AS product_3_revenue,
    SUM( CASE WHEN primary_product_id = 3 THEN price_usd - cogs_usd ELSE NULL END) AS product_3_margin,
    SUM( CASE WHEN primary_product_id = 4 THEN price_usd ELSE NULL END) AS product_4_revenue,
    SUM( CASE WHEN primary_product_id = 4 THEN price_usd - cogs_usd ELSE NULL END) AS product_4_margin
FROM ORDERS
WHERE created_at < '2015-03-20'
GROUP BY 1,2;
```

Output:

	yrs	mnth	total_revenue	total_margin	product_1_revenue	product_1_margin	product_2_revenue	product_2_margin	product_3_revenue	product_3_margin	product_4_revenue	product_4_margin
▶	2012	3	2999.40	1830.00	2999.40	1830.00	NULL	NULL	NULL	NULL	NULL	NULL
	2012	4	4949.01	3019.50	4949.01	3019.50	NULL	NULL	NULL	NULL	NULL	NULL
	2012	5	5398.92	3294.00	5398.92	3294.00	NULL	NULL	NULL	NULL	NULL	NULL
	2012	6	6998.60	4270.00	6998.60	4270.00	NULL	NULL	NULL	NULL	NULL	NULL
	2012	7	8448.31	5154.50	8448.31	5154.50	NULL	NULL	NULL	NULL	NULL	NULL
	2012	8	11397.72	6954.00	11397.72	6954.00	NULL	NULL	NULL	NULL	NULL	NULL
	2012	9	14347.13	8753.50	14347.13	8753.50	NULL	NULL	NULL	NULL	NULL	NULL
	2012	10	18546.29	11315.50	18546.29	11315.50	NULL	NULL	NULL	NULL	NULL	NULL
	2012	11	30893.82	18849.00	30893.82	18849.00	NULL	NULL	NULL	NULL	NULL	NULL
	2012	12	25294.94	15433.00	25294.94	15433.00	NULL	NULL	NULL	NULL	NULL	NULL
	2013	1	19966.10	12224.00	17146.57	10461.50	2819.53	1762.50	NULL	NULL	NULL	NULL
	2013	2	26515.02	16323.00	16796.64	10248.00	9718.38	6075.00	NULL	NULL	NULL	NULL
	2013	3	19896.15	12197.50	15996.80	9760.00	3899.35	2437.50	NULL	NULL	NULL	NULL
	2013	4	28584.47	17524.50	22945.41	13999.50	5639.06	3525.00	NULL	NULL	NULL	NULL
	2013	5	29364.29	17989.50	24445.11	14914.50	4919.18	3075.00	NULL	NULL	NULL	NULL
	2013	6	30544.07	18716.50	25144.97	15341.50	5399.10	3375.00	NULL	NULL	NULL	NULL
	2013	7	31143.96	19087.00	25444.91	15524.50	5699.05	3562.50	NULL	NULL	NULL	NULL
	2013	8	31373.92	19230.00	25494.90	15555.00	5879.02	3675.00	NULL	NULL	NULL	NULL
	2013	9	32723.65	20053.50	27094.59	16535.50	5629.06	3518.00	NULL	NULL	NULL	NULL
	2013	10	38242.62	23454.00	31943.67	19516.50	6298.95	3937.50	NULL	NULL	NULL	NULL
	2013	11	46631.02	28607.00	38192.43	23333.50	8438.59	5273.50	NULL	NULL	NULL	NULL
	2013	12	58262.60	36190.00	44063.23	27044.50	8980.49	5615.50	5218.88	3530.00	NULL	NULL
	2014	1	56568.89	35366.50	41049.81	25313.50	8948.49	5604.50	6570.59	4448.50	NULL	NULL
	2014	2	66012.52	41762.00	36586.22	22822.00	21997.99	13897.50	7428.31	5042.50	NULL	NULL
	2014	3	68189.73	43068.50	48611.65	30324.50	10718.05	6772.50	8860.03	5971.50	NULL	NULL
	2014	4	78725.43	49713.50	57631.72	35924.00	10666.08	6719.00	10427.63	7070.50	NULL	NULL
	2014	5	88935.27	56167.50	66531.80	41571.00	13125.59	8291.50	9277.88	6305.00	NULL	NULL
	2014	6	80051.25	50600.50	56839.92	35487.00	12611.72	7951.00	10599.61	7162.50	NULL	NULL
	2014	7	83288.55	52566.50	61260.97	38227.50	12023.83	7580.50	10003.75	6758.50	NULL	NULL
	2014	8	84716.08	53579.00	59611.27	37162.50	13125.57	8302.50	11979.24	8114.00	NULL	NULL
	2014	9	92232.49	58309.50	67437.53	42136.50	13817.45	8740.50	10977.51	7432.50	NULL	NULL
	2014	10	103905.98	65747.00	74440.01	46502.50	15319.22	9665.00	14146.75	9579.50	NULL	NULL
	2014	11	128162.98	80985.00	91530.42	57121.00	20494.27	12939.50	16138.29	10924.50	NULL	NULL
	2014	12	144823.02	91857.00	101328.51	63273.50	20042.32	12677.00	18355.81	12451.50	5096.38	3455.00
	2015	1	132211.54	83911.00	88219.21	55091.50	20926.10	13234.00	17058.15	11503.50	6008.08	4082.00
	2015	2	129212.94	82006.00	70456.93	44029.50	38958.95	24553.50	14642.71	9917.50	5154.35	3505.50
	2015	3	78951.07	50030.50	54248.48	33858.00	12013.79	7587.50	9171.89	6198.50	3516.91	2386.50

14. Assess the impact of introducing new products by tracking sessions to product pages and conversion rates over time.

Query:

```
WITH product_pageviews AS(
  SELECT
    website_session_id,
    website_pageview_id,
    created_at AS saw_product_page_at
  FROM website_pageviews
  WHERE pageview_url = '/products')
SELECT
  YEAR(pp.saw_product_page_at) AS yr,
  MONTH(pp.saw_product_page_at) AS mth,
  COUNT(DISTINCT pp.website_session_id) AS sessions_to_product_page,
  COUNT(DISTINCT wp.website_session_id) AS clicked_to_next_page,
  COUNT(DISTINCT wp.website_session_id)/COUNT(DISTINCT pp.website_session_id) AS clickthrough_rt,
  COUNT(o.order_id) AS orders,
  COUNT(o.order_id)/COUNT(DISTINCT pp.website_session_id) AS product_to_order_rt
FROM product_pageviews AS pp
LEFT JOIN website_pageviews AS wp
ON wp.website_session_id = pp.website_session_id
  AND wp.website_pageview_id > pp.website_pageview_id
LEFT JOIN orders AS o
ON o.website_session_id = pp.website_session_id
GROUP BY 1,2;
```

Output:

	yr	nth	sessions_to_product_page	clicked_to_next_page	clickthrough_rt	orders	product_to_order_rt
▶	2012	3	743	530	0.7133	300	0.4038
	2012	4	1447	1029	0.7111	495	0.3421
	2012	5	1584	1135	0.7165	540	0.3409
	2012	6	1752	1247	0.7118	700	0.3995
	2012	7	2018	1438	0.7126	845	0.4187
	2012	8	3012	2198	0.7297	1140	0.3785
	2012	9	3126	2258	0.7223	1435	0.4591
	2012	10	4030	2948	0.7315	1855	0.4603
	2012	11	6743	4849	0.7191	3090	0.4583
	2012	12	5013	3620	0.7221	2530	0.5047
	2013	1	3380	2595	0.7678	1955	0.5784
	2013	2	3685	2803	0.7607	2485	0.6744
	2013	3	3371	2576	0.7642	1925	0.5710
	2013	4	4362	3356	0.7694	2765	0.6339
	2013	5	4684	3609	0.7705	2855	0.6095
	2013	6	4600	3536	0.7687	2970	0.6457
	2013	7	5020	3890	0.7749	3015	0.6006
	2013	8	5226	3951	0.7560	3040	0.5817
	2013	9	5399	4072	0.7542	3145	0.5825
	2013	10	6038	4564	0.7559	3540	0.5863
	2013	11	7886	5900	0.7482	4305	0.5459
	2013	12	8840	7026	0.7948	5235	0.5922
	2014	1	7790	6387	0.8199	4915	0.6309
	2014	2	7960	6485	0.8147	5105	0.6413
	2014	3	8110	6669	0.8223	5325	0.6566
	2014	4	9744	7958	0.8167	6205	0.6368
	2014	5	10261	8465	0.8250	6840	0.6666
	2014	6	10011	8260	0.8251	6195	0.6188
	2014	7	10837	8958	0.8266	6435	0.5938
	2014	8	10768	8980	0.8340	6620	0.6148
	2014	9	11128	9156	0.8228	7120	0.6398
	2014	10	12335	10235	0.8298	8045	0.6522
	2014	11	14476	12020	0.8303	9925	0.6856
	2014	12	17240	14609	0.8474	11570	0.6711
	2015	1	15217	12992	0.8538	10495	0.6897
	2015	2	14373	12187	0.8479	10335	0.7191
	2015	3	9022	7723	0.8560	6270	0.6950

15. Evaluate cross-selling performance since the introduction of a new primary product in December 2014.

Query:

```
WITH data AS(
SELECT
    o.order_id,
    o.primary_product_id,
    o.created_at,
    oi.product_id,
    oi.is_primary_item
FROM orders AS o
LEFT JOIN order_items AS oi
ON o.order_id = oi.order_id
WHERE o.created_at > '2014-12-05')
SELECT
    primary_product_id,
    COUNT(DISTINCT order_id) AS total_orders,
    COUNT(CASE WHEN product_id = 1 AND is_primary_item = 0 THEN order_id ELSE NULL END) AS _xsold_p1,
    COUNT(CASE WHEN product_id = 2 AND is_primary_item = 0 THEN order_id ELSE NULL END) AS _xsold_p2,
    COUNT(CASE WHEN product_id = 3 AND is_primary_item = 0 THEN order_id ELSE NULL END) AS _xsold_p3,
    COUNT(CASE WHEN product_id = 4 AND is_primary_item = 0 THEN order_id ELSE NULL END) AS _xsold_p4,
    COUNT(CASE WHEN product_id = 1 AND is_primary_item = 0 THEN order_id ELSE NULL END)/COUNT(DISTINCT order_id) AS p1_xsell_rt,
    COUNT(CASE WHEN product_id = 2 AND is_primary_item = 0 THEN order_id ELSE NULL END)/COUNT(DISTINCT order_id) AS p2_xsell_rt,
    COUNT(CASE WHEN product_id = 3 AND is_primary_item = 0 THEN order_id ELSE NULL END)/COUNT(DISTINCT order_id) AS p3_xsell_rt,
    COUNT(CASE WHEN product_id = 4 AND is_primary_item = 0 THEN order_id ELSE NULL END)/COUNT(DISTINCT order_id) AS p4_xsell_rt
FROM data
GROUP BY 1
ORDER BY 1;
```

Output:

	primary_product_id	total_orders	_xsold_p1	_xsold_p2	_xsold_p3	_xsold_p4	p1_xsell_rt	p2_xsell_rt	p3_xsell_rt	p4_xsell_rt
▶	1	4467	0	238	553	933	0.0000	0.0533	0.1238	0.2089
	2	1277	25	0	40	260	0.0196	0.0000	0.0313	0.2036
	3	929	84	40	0	208	0.0904	0.0431	0.0000	0.2239
	4	581	16	9	22	0	0.0275	0.0155	0.0379	0.0000