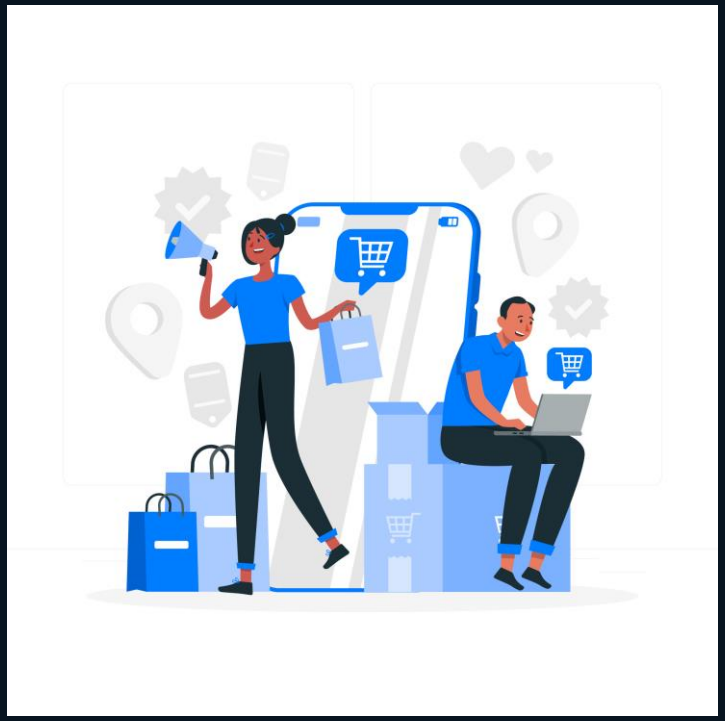
eCommerce Data Analyst Project – Insight Commerce

Project Background



Hired as an **eCommerce Data Analyst** for *Insight Commerce*, an online retailer.

Collaborated with: **CEO**, **Marketing Director**, **Website Manager**.

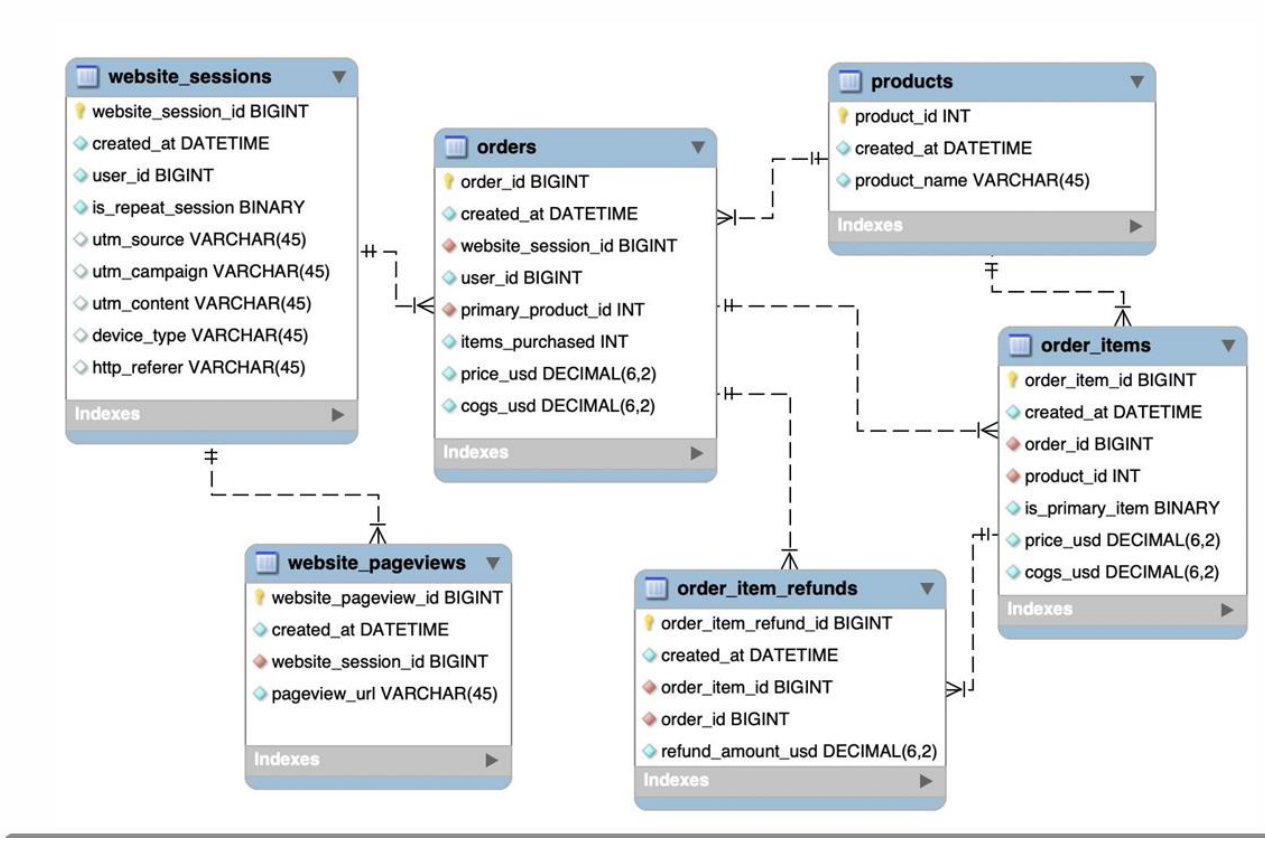
Key responsibilities:

Analyze & optimize **marketing channels**	Measure & test **website performance**	Evaluate **new product launches** impact using data

Project Overview

Project Phases

- The project was divided into two parts:
  - **Mid-Course Project**
  - **Final Project**



We will be working with six related tables, which contain eCommerce data about:

- **Website Activity**
- **Products**
- **Orders and Refunds**

We'll use MySQL to understand how customers access and interact with the site, analyze landing page performance and conversion, and explore product-level sales.

Database Schema Overview

Website Sessions Table

website_session_id	created_at	user_id	is_repeat_session	utm_source	utm_campaign	utm_content	device_type	http_referer
1	3/19/2012 8:04	1	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
2	3/19/2012 8:16	2	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
3	3/19/2012 8:26	3	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
4	3/19/2012 8:37	4	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
5	3/19/2012 9:00	5	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
6	3/19/2012 9:05	6	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
7	3/19/2012 9:06	7	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
8	3/19/2012 9:17	8	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
9	3/19/2012 9:27	9	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
10	3/19/2012 9:35	10	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
11	3/19/2012 9:37	11	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
12	3/19/2012 9:39	12	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
13	3/19/2012 9:45	13	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
14	3/19/2012 9:45	14	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
15	3/19/2012 9:57	15	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com

Products Table

product_id	created_at	product_name
1	3/19/2012 8:00	The Original Mr. Fuzzy
2	1/6/2013 13:00	The Forever Love Bear
3	12/12/2013 9:00	The Birthday Sugar Panda
4	2/5/2014 10:00	The Hudson River Mini bear

Website Pageviews Table

website_pageview_id	created_at	website_session_id	pageview_url
1	3/19/2012 8:04	1	/home
2	3/19/2012 8:16	2	/home
3	3/19/2012 8:26	3	/home
4	3/19/2012 8:37	4	/home
5	3/19/2012 9:00	5	/home
6	3/19/2012 9:05	6	/home
7	3/19/2012 9:06	7	/home
8	3/19/2012 9:10	6	/products
9	3/19/2012 9:10	6	/the-original-mr-fuzzy
10	3/19/2012 9:14	6	/cart
11	3/19/2012 9:16	6	/shipping
12	3/19/2012 9:17	8	/home

Orders Table

order_id	created_at	website_session_id	user_id	primary_product_id	items_purchased	price_usd	cogs_usd
1	3/19/2012 10:42	20	20	1	1	49.99	19.49
2	3/19/2012 19:27	104	104	1	1	49.99	19.49
3	3/20/2012 6:44	147	147	1	1	49.99	19.49
4	3/20/2012 9:41	160	160	1	1	49.99	19.49
5	3/20/2012 11:28	177	177	1	1	49.99	19.49
6	3/20/2012 16:12	232	232	1	1	49.99	19.49
7	3/20/2012 17:03	241	241	1	1	49.99	19.49
8	3/20/2012 23:35	295	295	1	1	49.99	19.49
9	3/21/2012 2:35	304	304	1	1	49.99	19.49
10	3/21/2012 6:45	317	317	1	1	49.99	19.49
11	3/21/2012 7:28	321	321	1	1	49.99	19.49
12	3/21/2012 8:21	327	327	1	1	49.99	19.49
13	3/21/2012 8:45	332	332	1	1	49.99	19.49
14	3/21/2012 9:12	336	336	1	1	49.99	19.49
15	3/21/2012 16:40	428	428	1	1	49.99	19.49

Order Items Table

order_item_id	created_at	order_id	product_id	is_primary_item	price_usd	cogs_usd
1	3/19/2012 10:42	1	1	1	49.99	19.49
2	3/19/2012 19:27	2	1	1	49.99	19.49
3	3/20/2012 6:44	3	1	1	49.99	19.49
4	3/20/2012 9:41	4	1	1	49.99	19.49
5	3/20/2012 11:28	5	1	1	49.99	19.49
6	3/20/2012 16:12	6	1	1	49.99	19.49
7	3/20/2012 17:03	7	1	1	49.99	19.49
8	3/20/2012 23:35	8	1	1	49.99	19.49
9	3/21/2012 2:35	9	1	1	49.99	19.49
10	3/21/2012 6:45	10	1	1	49.99	19.49
11	3/21/2012 7:28	11	1	1	49.99	19.49
12	3/21/2012 8:21	12	1	1	49.99	19.49
13	3/21/2012 8:45	13	1	1	49.99	19.49
14	3/21/2012 9:12	14	1	1	49.99	19.49
15	3/21/2012 16:40	15	1	1	49.99	19.49

Order Items Refund Table

order_item_refund_id	created_at	order_item_id	order_id	refund_amount_usd
1	4/6/2012 11:32	57	57	49.99
2	4/13/2012 1:09	74	74	49.99
3	4/15/2012 7:03	71	71	49.99
4	4/17/2012 20:00	118	118	49.99
5	4/22/2012 20:53	116	116	49.99
6	5/4/2012 11:59	147	147	49.99
7	5/12/2012 2:41	186	186	49.99
8	5/16/2012 13:06	191	191	49.99
9	5/24/2012 16:00	179	179	49.99
10	5/30/2012 17:20	199	199	49.99
11	6/6/2012 14:22	271	271	49.99
12	6/10/2012 19:54	290	290	49.99
13	6/20/2012 19:13	335	335	49.99
14	6/28/2012 18:54	382	382	49.99
15	6/29/2012 15:50	357	357	49.99

Data Dictionary

1. Website Sessions Table

Column Name	Description
website_session_id	Primary key – unique ID for each session
created_at	Timestamp of when the session occurred
user_id	User ID (linked to browser cookie) to track users across sessions
is_repeat_session	1 = returning visitor, 0 = first-time visitor
utm_source	Marketing source (e.g., gsearch, bsearch)
utm_campaign	Paid campaign identifier (nonbrand, brand, etc.)
utm_content	Specific ad name or variation (e.g., g_ad_1)
device_type	Device used to access the site (mobile, desktop)
http_referer	The referrer site from which user landed on the website

🟪 2. Website Pageviews Table

➡ This table tells us what pages were viewed and in what order during each session

Column Name	Description
website_pageview_id	Primary key – unique ID for each pageview
created_at	Timestamp of when the pageview occurred
website_session_id	Foreign key linking to the session the pageview belongs to
pageview_url	Page URL visited by the user

3. Orders Table

Column Name	Description
order_id	Primary key – unique order ID
created_at	Timestamp of when the order was placed
website_session_id	Foreign key linking to the session during which the order occurred
user_id	User who placed the order
primary_product_id	The first product user added to cart (used for attribution)
items_purchased	Number of items purchased
price_usd	Total sale amount (USD)
cogs_usd	Total cost of goods sold (USD)

4. Products Table

Column Name	Description
product_id	Unique product ID
created_at	Timestamp when the product was launched
product_name	Product's display name

5. Order Items Table

Column Name	Description
order_item_id	Unique ID for each item in every order
created_at	Timestamp of when item was ordered
order_id	Foreign key – which order this item belongs to
product_id	Product included in the order
is_primary_item	1 = this was the first item added to cart, 0 = cross-sell or secondary
price_usd	Price paid for this item
cogs_usd	Cost of this item

6. Order Item Refunds Table

Column Name	Description
order_item_refund_id	Unique ID for each refund
created_at	Timestamp of when refund occurred
order_item_id	Item that was refunded (foreign key to Order Items table)
order_id	Original order linked to the refund
refund_amount_usd	Refunded amount in USD

Key Analysis Performed

PROJECT DELIVERABLES

✅ TRAFFIC ANALYSIS & OPTIMIZATION

✅ WEBSITE MEASUREMENT & TESTING

✅ CHANNEL ANALYSIS & OPTIMIZATION

✅ PRODUCT-LEVEL ANALYSIS

✅ USER-LEVEL ANALYSIS

Mid course Project Objective

📌 *Insight Commerce has been live for ~8 months, and your CEO is due to present company performance metrics to the board next week. I need to prepare relevant metrics to show the company's promising growth.*

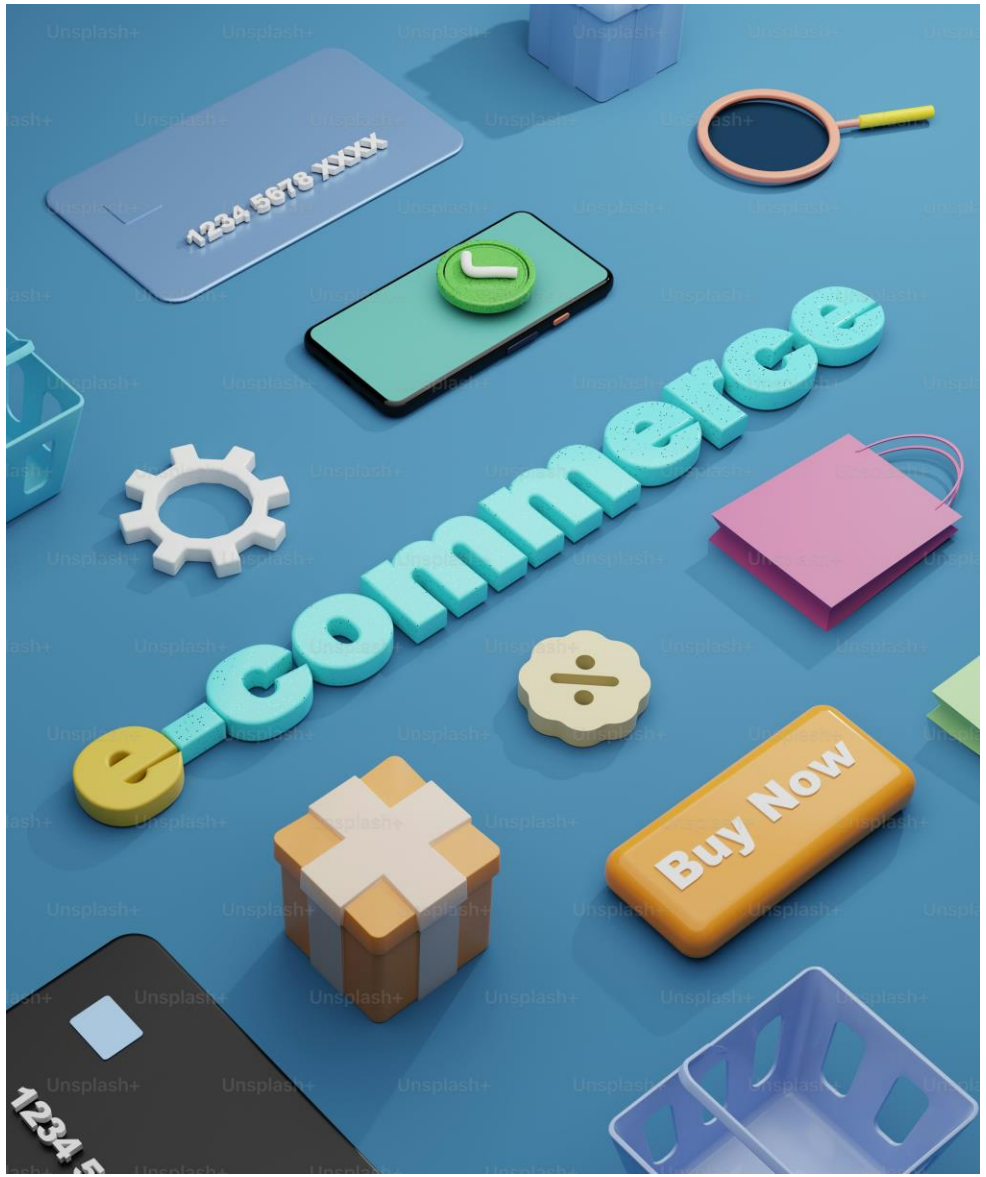
Tasks as an Analyst:

Use SQL to extract and analyze:

- **Website traffic**
- **Performance data**

Tell a compelling growth story using data

MID COURSE PROJECT QUESTIONS



Q.1 📈 How have monthly sessions and orders from Gsearch evolved over time?

Insights: **The results of this query show a clear upward trend in both sessions and orders originating from Gsearch, indicating the effectiveness of this channel.**

```sql
SELECT
    YEAR(w1.created_at) AS years,
    MONTH(w1.created_at) AS months,
    COUNT(DISTINCT w1.website_session_id) AS sessions,
    COUNT(DISTINCT o.order_id) AS orders,
    round(COUNT(DISTINCT o.order_id) * 100 / COUNT(DISTINCT w1.website_session_id),2) AS
conversion_rate
FROM website_sessions w1
    LEFT JOIN orders o
        ON o.website_session_id = w1.website_session_id
WHERE w1.created_at < '2012-11-27'
    AND w1.utm_source = 'gsearch'
GROUP BY 1,2;
```

years	months	sessions	orders	conversion_rate
2012	3	1860	60	3.23
2012	4	3574	92	2.57
2012	5	3410	97	2.84
2012	6	3578	121	3.38
2012	7	3811	145	3.80
2012	8	4877	184	3.77
2012	9	4491	188	4.19
2012	10	5534	234	4.23
2012	11	8889	373	4.20

Gsearch Performance Trends

Conversion Rate

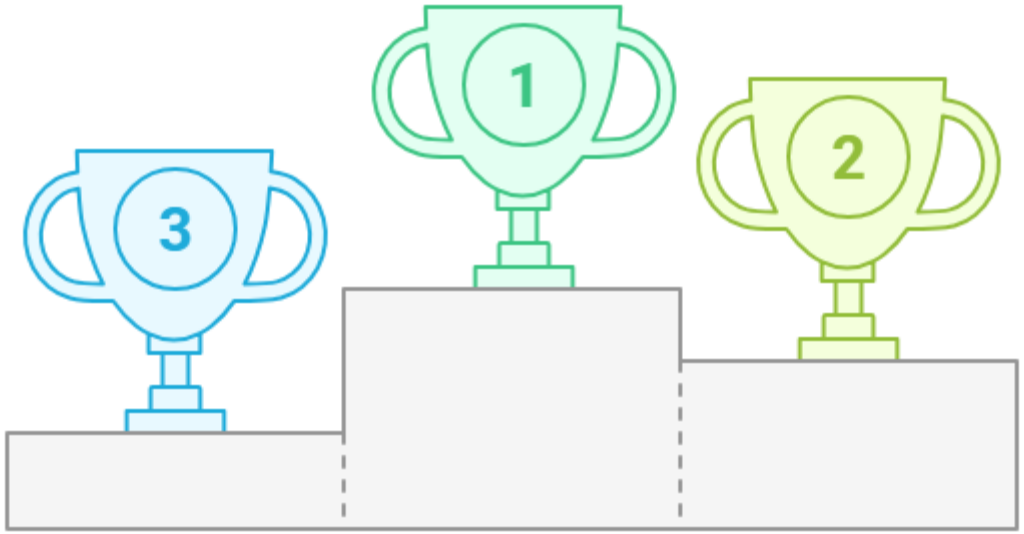
Conversion rate from Gsearch is steadily increasing.

Sessions

Sessions from Gsearch show a clear upward trend.

Orders

Orders from Gsearch also exhibit an upward trend.



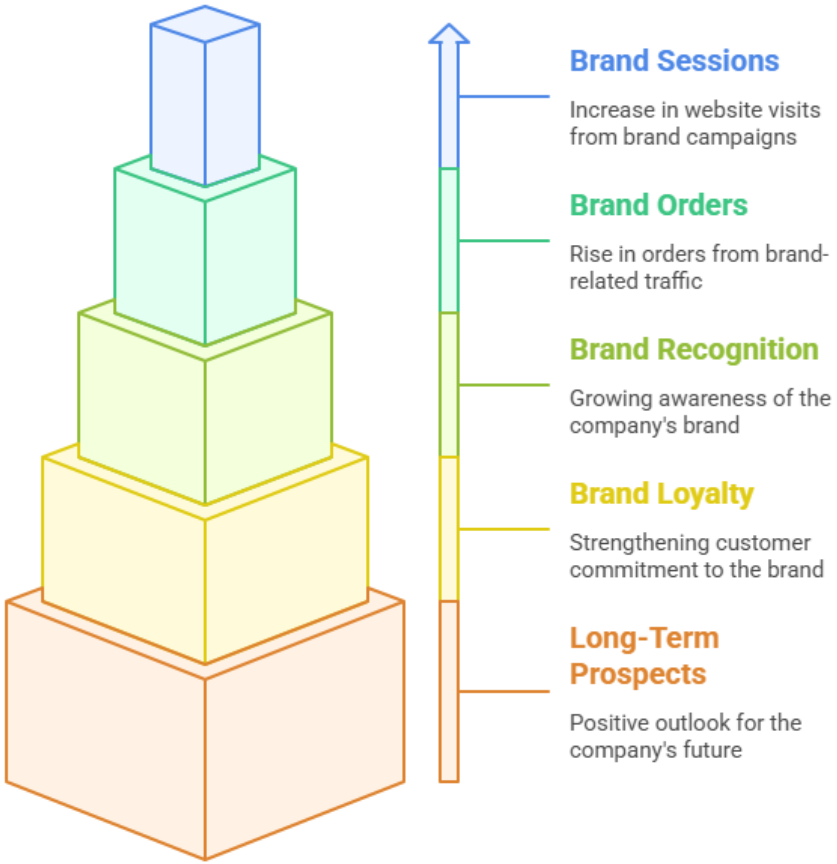
Q2. 🔍 Within Gsearch, how do brand and nonbrand campaigns compare in terms of monthly sessions and orders?

Insights: The data reveals a significant increase in both brand sessions and orders, suggesting that brand recognition and loyalty are growing. This is a positive sign for the company's long-term prospects.

```sql
SELECT
    YEAR(w1.created_at) AS years,
    MONTH(w1.created_at) AS months,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN w1.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 w1.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 w1
    LEFT JOIN orders o
        ON o.website_session_id = w1.website_session_id
WHERE w1.created_at < '2012-11-27'
    AND w1.utm_source = 'gsearch'
GROUP BY 1,2;
```

years	months	nonbrand_sessions	nonbrand_orders	brand_sessions	brand_orders
2012	3	1852	60	8	0
2012	4	3509	86	65	6
2012	5	3295	91	115	6
2012	6	3439	114	139	7
2012	7	3660	136	151	9
2012	8	4673	174	204	10
2012	9	4227	172	264	16
2012	10	5197	219	337	15
2012	11	8506	356	383	17

Brand Growth Pyramid



Brand Sessions

Increase in website visits from brand campaigns

Brand Orders

Rise in orders from brand-related traffic

Brand Recognition

Growing awareness of the company's brand

Brand Loyalty

Strengthening customer commitment to the brand

Long-Term Prospects

Positive outlook for the company's future

Q3. 📱 For Gsearch nonbrand traffic, how do sessions and orders vary by device type over time?

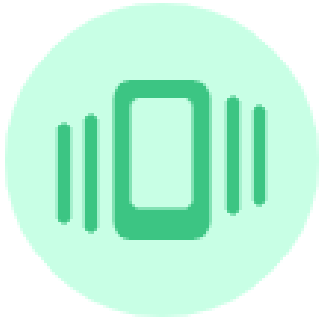
Insights: The results indicate that majority of traffic and orders from Gsearch nonbrand campaigns originate from desktop devices. This information can be used to optimize the website and marketing efforts for desktop users.

```sql
SELECT
    YEAR(w1.created_at) AS years,
    MONTH(w1.created_at) AS months,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN w1.website_session_id ELSE NULL END) AS
desktop_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN o.order_id ELSE NULL END) AS desktop_orders,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN w1.website_session_id ELSE NULL END) AS
mobile_sessions,
    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN o.order_id ELSE NULL END) AS mobile_orders
FROM website_sessions w1
    LEFT JOIN orders o
        ON o.website_session_id = w1.website_session_id
WHERE w1.created_at < '2012-11-27'
    AND w1.utm_source = 'gsearch'
    AND w1.utm_campaign = 'nonbrand'
GROUP BY 1,2;
```

	years	months	desktop_sessions	desktop_orders	mobile_sessions	mobile_orders
▶	2012	3	1128	50	724	10
	2012	4	2139	75	1370	11
	2012	5	2276	83	1019	8
	2012	6	2673	106	766	8
	2012	7	2774	122	886	14
	2012	8	3515	165	1158	9
	2012	9	3171	155	1056	17
	2012	10	3934	201	1263	18
	2012	11	6457	323	2049	33

Which device type should be prioritized for website optimization?

VS



Desktop

Optimize for desktop users

Mobile

Consider mobile users

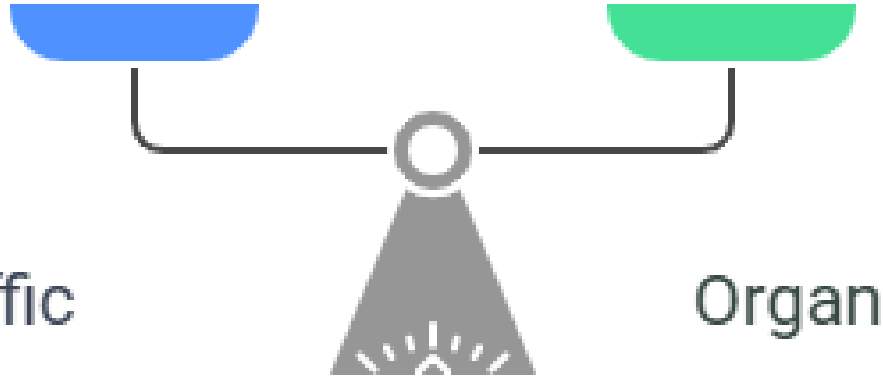
Q4. 🌐 **How does Gsearch traffic compare to other marketing channels in terms of monthly sessions and orders?**

Insights: **The data shows that while Gsearch is a significant contributor, organic search and direct type-in traffic are also growing, indicating a healthy diversification of traffic sources.**

```sql
SELECT DISTINCT
    utm_source,
    utm_campaign,
    http_referer
FROM website_sessions w1
WHERE w1.created_at < '2012-11-27';


SELECT
    YEAR(w1.created_at) AS yr,
    MONTH(w1.created_at) AS mo,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' THEN w1.website_session_id ELSE NULL END) AS
gsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' THEN w1.website_session_id ELSE NULL END) AS
bsearch_paid_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN w1.website_session_id
ELSE NULL END)
    AS organic_search_sessions,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN w1.website_session_id
ELSE NULL END)
    AS direct_type_in_sessions
FROM website_sessions w1
     LEFT JOIN orders o
        ON o.website_session_id = w1.website_session_id
WHERE w1.created_at < '2012-11-27'
GROUP BY 1,2;
```

yr	mo	gsearch_paid_sessions	bsearch_paid_sessions	organic_search_sessions	direct_type_in_sessions
2012	3	1860	2	8	9
2012	4	3574	11	78	71
2012	5	3410	25	150	151
2012	6	3578	25	190	170
2012	7	3811	44	207	187
2012	8	4877	705	265	250
2012	9	4491	1439	331	285
2012	10	5534	1781	428	440
2012	11	8889	2840	536	485

Significant traffic contributor



Growing traffic sources

Gsearch Traffic

Organic/Direct Traffic

Q5.🧪 How has our website's session-to-order conversion rate changed month-over-month?

Insights: **The results show a conversion rate consistently above 4%, indicating effective website design and user experience.**

```sql
SELECT
    YEAR(w1.created_at) AS yr,
    MONTH(w1.created_at) AS mo,
    COUNT(DISTINCT w1.website_session_id) AS sessions,
    COUNT(DISTINCT o.order_id) AS orders,
    round(COUNT(DISTINCT o.order_id)*100/COUNT(DISTINCT w1.website_session_id),2) AS conversion_rate

FROM website_sessions w1
    LEFT JOIN orders o
        ON o.website_session_id = w1.website_session_id
WHERE w1.created_at < '2012-11-27'
GROUP BY 1,2;
```

	yr	mo	sessions	orders	conversion_rate
▶	2012	3	1879	60	3.19
	2012	4	3734	99	2.65
	2012	5	3736	108	2.89
	2012	6	3963	140	3.53
	2012	7	4249	169	3.98
	2012	8	6097	228	3.74
	2012	9	6546	287	4.38
	2012	10	8183	371	4.53
	2012	11	12750	561	4.40

Conversion Rate Success Pyramid

Conversion Rate

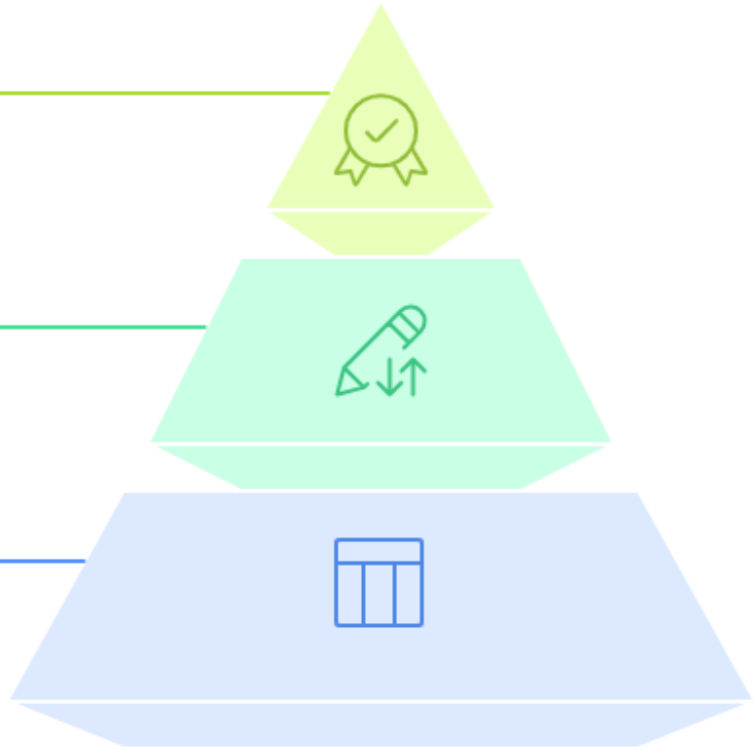
Consistently above 4%, indicating success

User Experience

Positive interaction leads to higher conversions

Effective Design

Website design enhances user engagement



Q6. 💰 **What was the estimated revenue impact of the Gsearch landing page test conducted between Jun 19 and Jul 28?**

Insights : **The analysis concludes that the lander test resulted in an estimated 200 incremental orders since the test began, translating to roughly 50 extra orders per month. And 22,972 website sessions since the test.**

```sql
SELECT
    MIN(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url = '/lander-1';



-- for this step, we'll find the first pageview id

CREATE TEMPORARY TABLE first_test_pageviews
SELECT
    w2.website_session_id,
    MIN(w2.website_pageview_id) AS first_pageview_id
FROM website_pageviews w2
    INNER JOIN website_sessions w1
        ON w1.website_session_id = w2.website_session_id
        AND w1.created_at < '2012-07-28'
        AND w2.website_pageview_id ≥ 23504 -- first page_view
        AND utm_source = 'gsearch'
        AND utm_campaign = 'nonbrand'
GROUP BY
    w2.website_session_id;
```

```sql
-- next, we'll bring in the landing page to each session but restricting to home or lander-1 this time

CREATE TEMPORARY TABLE nonbrand_test_sessions_w_landing_pages

SELECT
    f.website_session_id,
    w2.pageview_url AS landing_page
FROM first_test_pageviews f
    LEFT JOIN website_pageviews w2
        ON w2.website_pageview_id = f.first_pageview_id
WHERE w2.pageview_url IN ('/home','/lander-1');

SELECT * FROM nonbrand_test_sessions_w_landing_pages;

-- then we make a table to bring in orders

CREATE TEMPORARY TABLE nonbrand_test_sessions_w_orders
SELECT
    n1.website_session_id,
    n1.landing_page,
    o.order_id AS order_id
FROM nonbrand_test_sessions_w_landing_pages n1
LEFT JOIN orders o
    ON o.website_session_id = n1.website_session_id;


SELECT * FROM nonbrand_test_sessions_w_orders;

-- to find the difference between conversion rates
SELECT
    landing_page,
    COUNT(DISTINCT website_session_id) AS sessions,
    COUNT(DISTINCT order_id) AS orders,
    COUNT(DISTINCT order_id)*100/COUNT(DISTINCT website_session_id) AS conv_rate
FROM nonbrand_test_sessions_w_orders
GROUP BY 1;

-- .0319 for /home, vs .0406 for /lander-1
-- .0087 additional orders per session
```

```sql
-- to find the difference between conversion rates
SELECT
    landing_page,
    COUNT(DISTINCT website_session_id) AS sessions,
    COUNT(DISTINCT order_id) AS orders,
    COUNT(DISTINCT order_id)*100/COUNT(DISTINCT website_session_id) AS conv_rate
FROM nonbrand_test_sessions_w_orders
GROUP BY 1;

-- .0319 for /home, vs .0406 for /lander-1
-- .0087 additional orders per session
-- finding the most recent pageview for gsearch nonbrand where the traffic was sent to /home

SELECT
    MAX(w1.website_session_id) AS most_recent_gsearch_nonbrand_home_pageview
FROM website_sessions w1
    LEFT JOIN website_pageviews w2
        ON w1.website_session_id = w2.website_session_id
WHERE utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand'
    AND pageview_url = '/home'
    AND w1.created_at < '2012-11-27';
-- max website_session_id = 17145


SELECT
    COUNT(website_session_id) AS sessions_since_test
FROM website_sessions
WHERE created_at < '2012-11-27'
    AND website_session_id > 17145 -- last /home session
    AND utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand';

-- 22,972 website sessions since the test
-- .0087 incremental conversion = 202 incremental orders since 7/29 (200 orders approx)
-- roughly 4 months, so roughly 50 extra orders per month.
```

	landing_page	sessions	orders	conv_rate
▶	/home	2261	72	3.1844
	/lander-1	2316	94	4.0587

	sessions_since_test
▶	22972

Gsearch Lander Test Revenue Estimation

Website Sessions

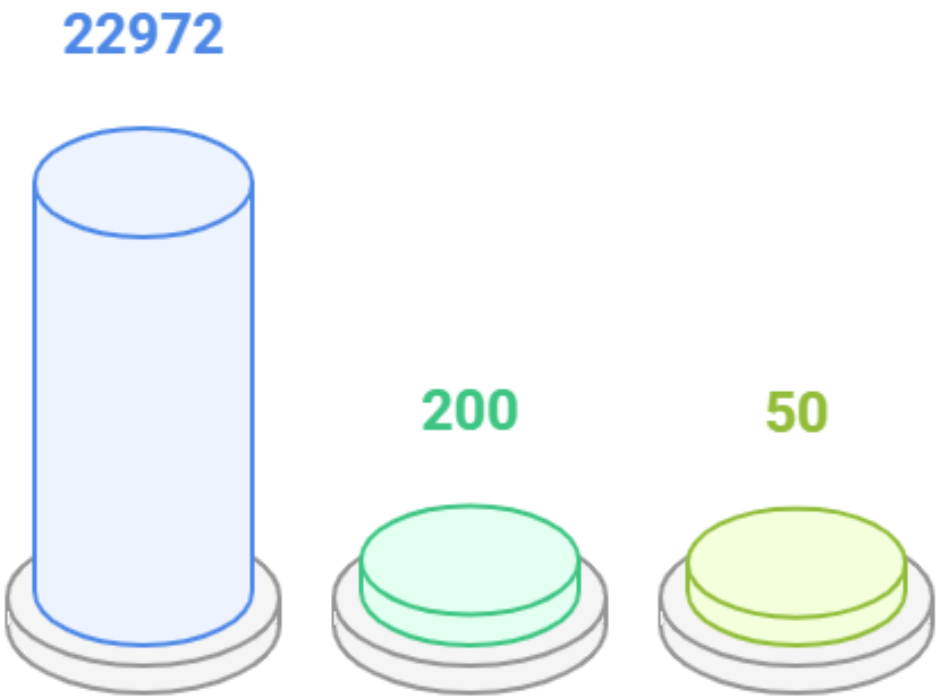
Total sessions since test began

Incremental Orders

Additional orders since test start

Monthly Extra Orders

Average extra orders per month



Q7. 🔄 **What does the full conversion funnel look like for each of the two landing pages tested during Jun 19- Jul 28?**

Insights: Custom Lander users progress more through checkout
Shipping click rate is higher for custom lander (85.28%) vs. homepage (84.00%)
Billing click rate is also higher (47.72% vs. 42.86%) – indicating stronger final funnel progression among custom lander users.

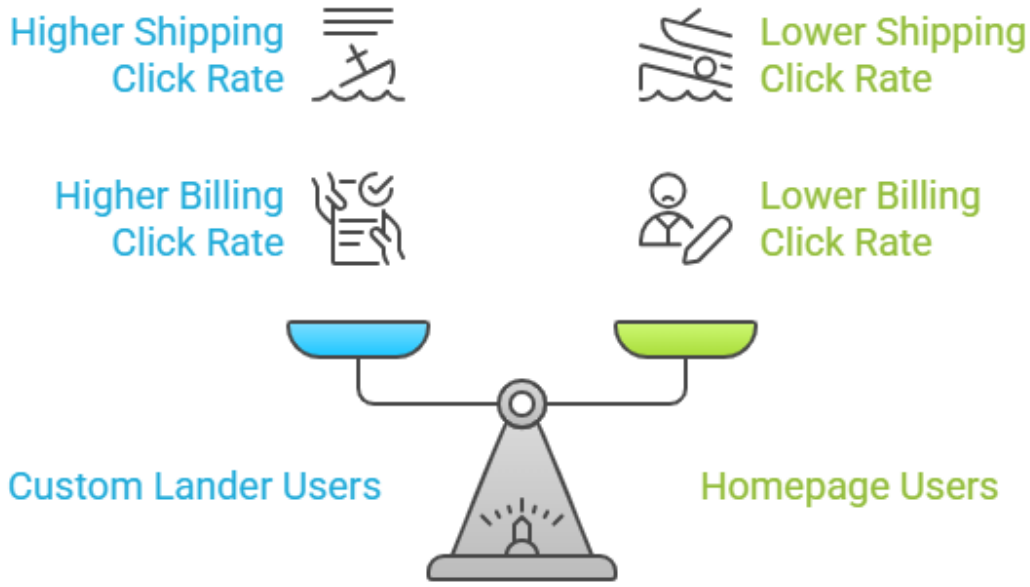
```sql
SELECT
    w1.website_session_id,
    w2.pageview_url,
    -- website_pageviews.created_at AS pageview_created_at,
    CASE WHEN pageview_url = '/home' THEN 1 ELSE 0 END AS homepage,
    CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS custom_lander,
    CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END AS products_page,
    CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS mrfuzzy_page,
    CASE WHEN pageview_url = '/cart' THEN 1 ELSE 0 END AS cart_page,
    CASE WHEN pageview_url = '/shipping' THEN 1 ELSE 0 END AS shipping_page,
    CASE WHEN pageview_url = '/billing' THEN 1 ELSE 0 END AS billing_page,
    CASE WHEN pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS thankyou_page
FROM website_sessions w1
    LEFT JOIN website_pageviews  w2
        ON w1.website_session_id = w2.website_session_id
WHERE w1.utm_source = 'gsearch'
    AND w1.utm_campaign = 'nonbrand'
    AND w1.created_at < '2012-07-28'
        AND w1.created_at > '2012-06-19'
ORDER BY
    w1.website_session_id,
    w2.created_at;
```

```sql
-- then this would produce the final output, part 1

SELECT
    CASE
        WHEN saw_homepage = 1 THEN 'saw_homepage'
        WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
        ELSE 'uh oh ... check logic'
    END AS segment,
    COUNT(DISTINCT website_session_id) AS sessions,
    COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL END) AS to_products,
    COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL END) AS to_mrfuzzy,
    COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END) AS to_cart,
    COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN website_session_id ELSE NULL END) AS
to_shipping,
    COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN website_session_id ELSE NULL END) AS to_billing,
    COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN website_session_id ELSE NULL END) AS to_thankyou
FROM session_level_made_it_flagged
GROUP BY 1;



-- then this as final output part 2 - click rates

SELECT
    CASE
        WHEN saw_homepage = 1 THEN 'saw_homepage'
        WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
        ELSE 'uh oh ... check logic'
    END AS segment,
    COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL
END)*100/COUNT(DISTINCT website_session_id) AS lander_click_rt,
    COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL
END)*100/COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL END) AS
products_click_rt,
    COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END)*100/COUNT(DISTINCT
CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL END) AS mrfuzzy_click_rt,
    COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN website_session_id ELSE NULL
END)*100/COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END) AS
cart_click_rt,
    COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN website_session_id ELSE NULL
END)*100/COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN website_session_id ELSE NULL END) AS
shipping_click_rt,
    COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN website_session_id ELSE NULL
END)*100/COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN website_session_id ELSE NULL END) AS
billing_click_rt
FROM session_level_made_it_flagged
GROUP BY 1;
```

```sql
CREATE TEMPORARY TABLE session_level_made_it_flagged
SELECT
    website_session_id,
    MAX(homepage) AS saw_homepage,
    MAX(custom_lander) AS saw_custom_lander,
    MAX(products_page) AS product_made_it,
    MAX(mrfuzzy_page) AS mrfuzzy_made_it,
    MAX(cart_page) AS cart_made_it,
    MAX(shipping_page) AS shipping_made_it,
    MAX(billing_page) AS billing_made_it,
    MAX(thankyou_page) AS thankyou_made_it
FROM(
SELECT
    w1.website_session_id,
    w2.pageview_url,
    -- website_pageviews.created_at AS pageview_created_at,
    CASE WHEN pageview_url = '/home' THEN 1 ELSE 0 END AS homepage,
    CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS custom_lander,
    CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END AS products_page,
    CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS mrfuzzy_page,
    CASE WHEN pageview_url = '/cart' THEN 1 ELSE 0 END AS cart_page,
    CASE WHEN pageview_url = '/shipping' THEN 1 ELSE 0 END AS shipping_page,
    CASE WHEN pageview_url = '/billing' THEN 1 ELSE 0 END AS billing_page,
    CASE WHEN pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS thankyou_page
FROM website_sessions w1
    LEFT JOIN website_pageviews  w2
        ON w1.website_session_id = w2.website_session_id
WHERE w1.utm_source = 'gsearch'
    AND w1.utm_campaign = 'nonbrand'
    AND w1.created_at < '2012-07-28'
        AND w1.created_at > '2012-06-19'
ORDER BY
    w1.website_session_id,
    w2.created_at
) AS pageview_level
GROUP BY
    website_session_id;
```

segment	lander_click_rt	products_click_rt	mrfuzzy_click_rt	cart_click_rt	shipping_click_rt	billing_click_rt
saw_custom_lander	46.7617	71.2835	45.0777	66.3793	85.2814	47.7157
saw_homepage	41.6630	72.6115	43.2749	67.5676	84.0000	42.8571

Custom Lander Users Show Stronger Funnel Progression



Q.8. 📜 **What was the revenue lift from the billing page test (Sep 10 – Nov 10), and what is its estimated monthly impact?**

Insights: - **$22.83 revenue per billing page seen for the old version, $31.34 for the new version, LIFT: $8.51 per billing session, 1,193 billing sessions past month, VALUE OF BILLING TEST: $10,152 over the past month (1193*8.51)**

```sql
SELECT
    w2.website_session_id,
    w2.pageview_url AS billing_version_seen,
    o.order_id,
    o.price_usd
FROM website_pageviews w2
    LEFT JOIN orders o
        ON o.website_session_id = w2.website_session_id
WHERE w2.created_at > '2012-09-10'
    AND w2.created_at < '2012-11-10'
    AND w2.pageview_url IN ('/billing','/billing-2');
```

```sql
-- Revenue per billing page session

SELECT
    billing_version_seen,
    COUNT(DISTINCT website_session_id) AS sessions,
    round(SUM(price_usd)/COUNT(DISTINCT website_session_id) ,2)
    AS revenue_per_billing_page_seen
 FROM( SELECT
    w2.website_session_id,
    w2.pageview_url AS billing_version_seen,
    o.order_id,
    o.price_usd
FROM website_pageviews w2
    LEFT JOIN orders o
        ON o.website_session_id = w2.website_session_id
WHERE w2.created_at > '2012-09-10'
    AND w2.created_at < '2012-11-10'
    AND w2.pageview_url IN ('/billing','/billing-2')
    ) as billing_pageviews_and_order_data
    GROUP BY 1;


-- $22.83 revenue per billing page seen for the old version
-- $31.34 for the new version
-- LIFT: $8.51 per billing page view

-- the number of billing page sessions for the past month

SELECT
    COUNT(website_session_id) AS billing_sessions_past_month
FROM website_pageviews
WHERE website_pageviews.pageview_url IN ('/billing','/billing-2')
    AND created_at BETWEEN '2012-10-27' AND '2012-11-27' ;-- past
month
-- 1,193 billing sessions past month
-- LIFT: $8.51 per billing session
-- VALUE OF BILLING TEST: $10,152 over the past month (1193*8.51)
```

billing_version_seen	sessions	revenue_per_billing_page_seen
/billing	657	22.826484
/billing-2	654	31.339297

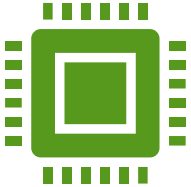
billing_sessions_past_month
1193

Billing Version Comparison

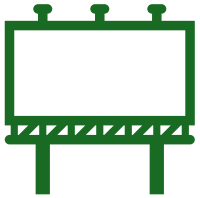
Metric	Old Version	New Version
Revenue per page	$22.83	$31.34
LIFT per session	N/A	$8.51
Sessions past month	N/A	1,193
Value past month	N/A	$10,152

Final Project Objective

Assist CEO (Cindy) in presenting to investors to secure next round of funding



Show *Insight Commerce* as a data-driven, fast-growing company



Highlight:

Key marketing channel activities

Website optimizations

Product and user-level insights

FINAL COURSE PROJECT QUESTIONS

Q1. 📊 How have overall sessions and orders trended quarterly since launch?

```sql
SELECT
    YEAR(w.created_at) AS yr,
    QUARTER(w.created_at) AS qtr,
    COUNT(DISTINCT w.website_session_id) AS sessions,
    COUNT(DISTINCT o.order_id) AS orders
FROM website_sessions w
    LEFT JOIN orders o
        ON w.website_session_id = o.website_session_id
GROUP BY 1,2
ORDER BY 1,2;
```

	yr	qtr	sessions	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

Growth of Sessions and Orders

Initial Slow Growth

Sessions and orders start with slow momentum

Significant Pickup

Sessions and orders begin to increase noticeably

Steady Growth

Sessions and orders continue to grow consistently

Peak Order Volume

Orders reach their highest point in 2014

Insights:

- Steady growth in sessions and orders from Q1 2012 to Q4 2014.
- Initial traction was slow but showed significant pickup starting Q3 2012.
- Peak order volume observed in 2014, indicating effective scaling.

Recommendations:

- ✅ Maintain marketing investment levels that fueled 2013–2014 growth.
- ✅ Capitalize on Q4 Demand:
- ✅ Increase inventory and marketing spend before holiday surges.
- ✅ Launch early Black Friday promotions.
- ✅ Sustain Growth:
- ✅ Invest in server capacity to handle peak traffic.
- ✅ Optimize checkout flow to prevent cart abandonment during high-volume periods.

Q2. ⚙️ What improvements have we made quarterly, in session-to-order conversion rate, revenue per order, and revenue per session over time?

```sql
SELECT
    YEAR(w.created_at) AS yr,
    QUARTER(w.created_at) AS qtr,
    round(COUNT(DISTINCT o.order_id)*100/COUNT(DISTINCT w.website_session_id),2) AS
session_to_order_conv_rate,
    round(SUM(price_usd)/COUNT(DISTINCT o.order_id),2) AS revenue_per_order,
    round(SUM(price_usd)/COUNT(DISTINCT w.website_session_id),2) AS revenue_per_session
FROM website_sessions w
    LEFT JOIN orders o
        ON w.website_session_id = o.website_session_id
GROUP BY 1,2
ORDER BY 1,2;
ORDER BY 1,2;
```

yr	qtr	session_to_order_conv_rate	revenue_per_order	revenue_per_session
2012	1	3.19	49.99	1.60
2012	2	3.04	49.99	1.52
2012	3	4.05	49.99	2.02
2012	4	4.63	49.99	2.32
2013	1	6.42	52.14	3.35
2013	2	6.94	51.54	3.58
2013	3	6.65	51.73	3.44
2013	4	6.45	54.72	3.53
2014	1	6.56	62.16	4.08
2014	2	7.24	64.37	4.66
2014	3	7.06	64.49	4.55
2014	4	7.74	63.79	4.93
2015	1	8.44	62.80	5.30

Insights:

- Conversion rates rose from 3.2% in early 2012 to ~8.5%+ by late 2014.
- Revenue per order remained stable with slight increases.
- Revenue per session consistently improved—driven by better targeting and conversion.

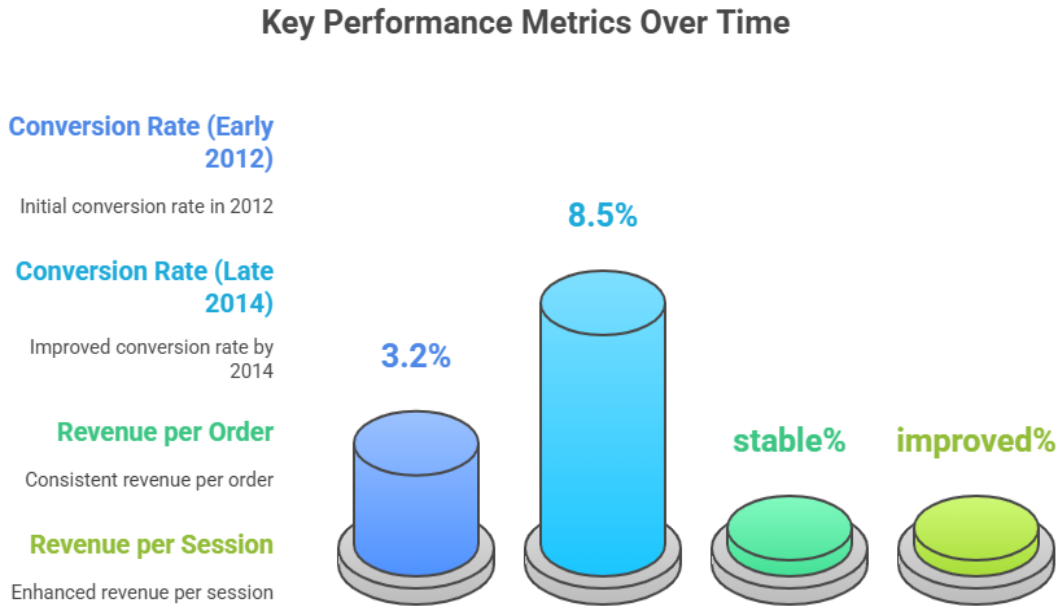
Recommendations:

✅ Highlight key conversion optimization campaigns (A/B tests, UX revamp, etc.) for internal learning and reuse. Focus on optimizing the user experience and checkout process to further enhance conversion rates.

✅ Explore advanced personalization or retargeting to further boost conversion and revenue per session. Continue to invest in marketing and sales channels to maintain and accelerate this growth momentum.

✅ Test Price Elasticity:

Gradually increase prices on bestsellers (customers tolerate higher prices).



Q3. 📣 How have orders from key channels (Gsearch Nonbrand, Bsearch Nonbrand, Brand, Organic, Direct) grown quarterly?

```sql
SELECT
    YEAR(w.created_at) AS yr,
    QUARTER(w.created_at) AS qtr,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN o.order_id ELSE
NULL END) AS gsearch_nonbrand_orders,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN o.order_id ELSE
NULL END) AS bsearch_nonbrand_orders,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN o.order_id ELSE NULL END) AS
brand_search_orders,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN o.order_id ELSE NULL
END) AS organic_search_orders,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN o.order_id ELSE NULL END)
AS direct_type_in_orders

FROM website_sessions w
    LEFT JOIN orders o
        ON w.website_session_id = o.website_session_id
GROUP BY 1,2
ORDER BY 1,2;
```

yr	qtr	gsearch_nonbrand_orders	bsearch_nonbrand_orders	brand_search_orders	organic_search_orders	direct_type_in_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

Insights:

- Gsearch non-brand was the dominant early driver of orders.
- Brand search and direct visits increased significantly over time is a strong indicator of brand equity growth.
- Direct and Organic search also gained traction, showing good SEO practices.
- The business has become much less dependent on gsearch nonbrand campaign , and it has starting to build its own brand. Organic, Direct type in traffic which has better margin and it takes you out of the dependency search engine.

Recommendations:

✅ Invest in long-term brand campaigns (offline + online).
✅ Scale SEO efforts to reduce dependency on paid channels.
✅ Leverage high-performing keywords in Bsearch and Gsearch to improve performance
✅ Invest in email marketing & retargeting to drive repeat purchases.

Q4. 📈 How have conversion rates for each of those channels evolved, and if any period where did we see major improvements?

```sql
SELECT
    YEAR(w.created_at) AS yr,
    QUARTER(w.created_at) AS qtr,
   round( COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN o.order_id
ELSE NULL END)*100
        /COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN
w.website_session_id ELSE NULL END),2) AS gsearch_nonbrand_conv_rt,

   round( COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN o.order_id
ELSE NULL END) *100
        /COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN
w.website_session_id ELSE NULL END),2) AS bsearch_nonbrand_conv_rt,

   round( COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN o.order_id ELSE NULL END)*100
        /COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN w.website_session_id ELSE NULL END),2) AS
brand_search_conv_rt,

   round( COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN o.order_id ELSE
NULL END) *100
        /COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN
w.website_session_id ELSE NULL END),2) AS organic_search_conv_rt,

    round(COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN o.order_id ELSE NULL
END) *100
        /COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN w.website_session_id
ELSE NULL END),2) AS direct_type_in_conv_rt
      FROM website_sessions w
    LEFT JOIN orders o
        ON w.website_session_id = o.website_session_id
GROUP BY 1,2
ORDER BY 1,2;
```

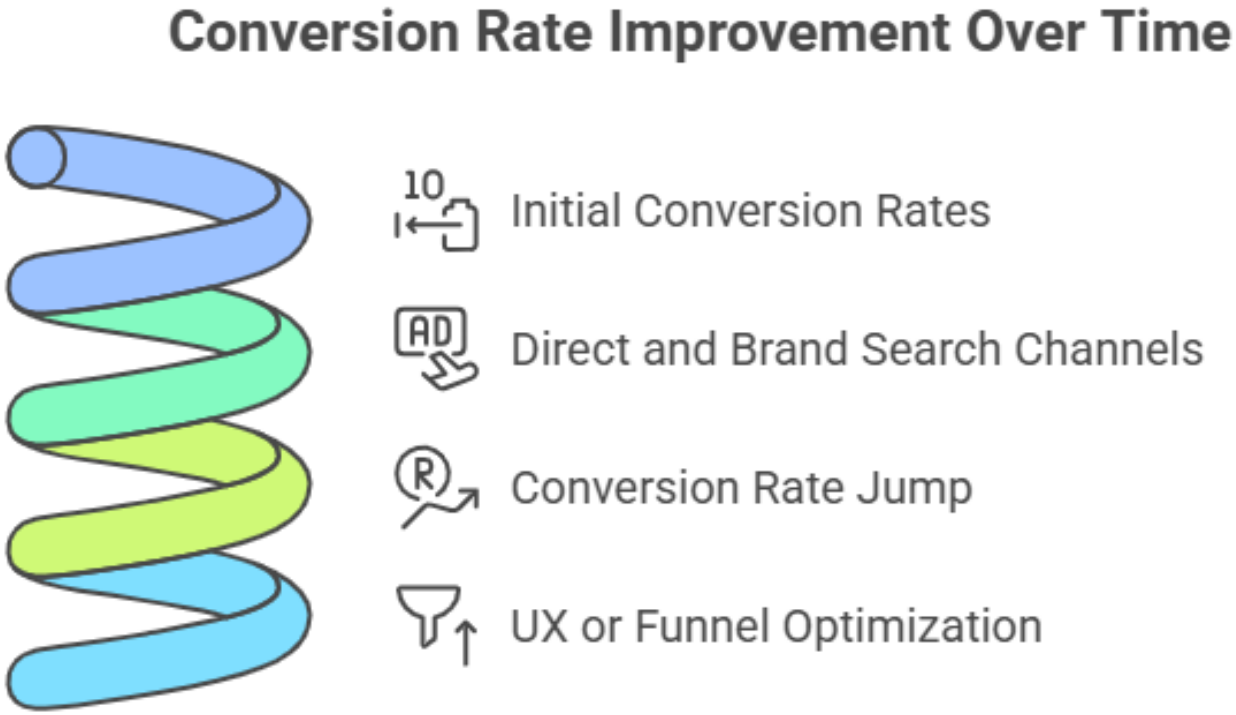
yr	qtr	gsearch_nonbrand_conv_rt	bsearch_nonbrand_conv_rt	brand_search_conv_rt	organic_search_conv_rt	direct_type_in_conv_rt
2012	1	3.24	NULL	0.00	0.00	0.00
2012	2	2.84	NULL	5.26	3.59	5.36
2012	3	3.84	4.08	6.02	4.98	4.43
2012	4	4.36	4.97	5.31	5.39	5.37
2013	1	6.12	6.93	7.03	7.53	6.14
2013	2	6.85	6.90	6.79	7.60	7.35
2013	3	6.39	6.97	7.03	7.34	7.19
2013	4	6.29	6.01	8.01	6.94	6.47
2014	1	6.93	7.04	8.39	7.56	7.65
2014	2	7.02	6.95	8.04	7.97	7.38
2014	3	7.03	6.98	7.56	7.33	7.02
2014	4	7.82	8.41	8.12	7.84	7.48
2015	1	8.61	8.50	8.52	8.21	7.75

Insights:

- Conversion rates improved for almost all channels over time.
- Direct and brand search channels had consistently higher conversion rates—again highlighting brand strength.
- Notable jump in conversion across channels around 2013–2014 suggests major UX or funnel optimization.

Recommendations:

✅ Double down on brand-building to increase direct and brand search traffic.
✅ Identify and replicate successful optimizations done around 2013-2014.
✅ Improve weaker channels (e.g., Bsearch) with tighter targeting.
Like Improving Bsearch landing pages to match Gsearch performance.



Q5. 🛍️ How have revenue and margin trended monthly by product along with Total sales and Revenue, and what seasonal patterns can we observe?

```sql
SELECT
    YEAR(created_at) AS yr,
    MONTH(created_at) AS mo,
    SUM(CASE WHEN product_id = 1 THEN price_usd ELSE NULL END) AS mrfuzzy_rev,
    SUM(CASE WHEN product_id = 1 THEN price_usd - cogs_usd ELSE NULL END) AS mrfuzzy_marg,
    SUM(CASE WHEN product_id = 2 THEN price_usd ELSE NULL END) AS lovebear_rev,
    SUM(CASE WHEN product_id = 2 THEN price_usd - cogs_usd ELSE NULL END) AS lovebear_marg,
    SUM(CASE WHEN product_id = 3 THEN price_usd ELSE NULL END) AS birthdaybear_rev,
    SUM(CASE WHEN product_id = 3 THEN price_usd - cogs_usd ELSE NULL END) AS birthdaybear_marg,
    SUM(CASE WHEN product_id = 4 THEN price_usd ELSE NULL END) AS minibear_rev,
    SUM(CASE WHEN product_id = 4 THEN price_usd - cogs_usd ELSE NULL END) AS minibear_marg,
    SUM(price_usd) AS total_revenue,
    SUM(price_usd - cogs_usd) AS total_margin
FROM order_items
GROUP BY 1,2
ORDER BY 1,2;
```

yr	mo	mrfuzzy_rev	mrfuzzy_marg	lovebear_rev	lovebear_marg	birthdaybear_rev	birthdaybear_marg	minibear_rev	minibear_marg	total_revenue	total_margin
2012	3	2999.40	1830.00	NULL	NULL	NULL	NULL	NULL	NULL	2999.40	1830.00
2012	4	4949.01	3019.50	NULL	NULL	NULL	NULL	NULL	NULL	4949.01	3019.50
2012	5	5398.92	3294.00	NULL	NULL	NULL	NULL	NULL	NULL	5398.92	3294.00
2012	6	6998.60	4270.00	NULL	NULL	NULL	NULL	NULL	NULL	6998.60	4270.00
2012	7	8448.31	5154.50	NULL	NULL	NULL	NULL	NULL	NULL	8448.31	5154.50
2012	8	11397.72	6954.00	NULL	NULL	NULL	NULL	NULL	NULL	11397.72	6954.00
2012	9	14347.13	8753.50	NULL	NULL	NULL	NULL	NULL	NULL	14347.13	8753.50
2012	10	18546.29	11315.50	NULL	NULL	NULL	NULL	NULL	NULL	18546.29	11315.50
2012	11	30893.82	18849.00	NULL	NULL	NULL	NULL	NULL	NULL	30893.82	18849.00
2012	12	25294.94	15433.00	NULL	NULL	NULL	NULL	NULL	NULL	25294.94	15433.00
2013	1	17146.57	10461.50	2819.53	1762.50	NULL	NULL	NULL	NULL	19966.10	12224.00
2013	2	16796.64	10248.00	9718.38	6075.00	NULL	NULL	NULL	NULL	26515.02	16323.00
2013	3	15996.80	9760.00	3899.35	2437.50	NULL	NULL	NULL	NULL	19896.15	12197.50
2013	4	22945.41	13999.50	5639.06	3525.00	NULL	NULL	NULL	NULL	28584.47	17524.50
2013	5	24445.11	14914.50	4919.18	3075.00	NULL	NULL	NULL	NULL	29364.29	17989.50
2013	6	25144.97	15341.50	5399.10	3375.00	NULL	NULL	NULL	NULL	30544.07	18716.50
2013	7	25444.91	15524.50	5699.05	3562.50	NULL	NULL	NULL	NULL	31143.96	19087.00
2013	8	25494.90	15555.00	5879.02	3675.00	NULL	NULL	NULL	NULL	31373.92	19230.00

yr	mo	mrfuzzy_rev	mrfuzzy_marg	lovebear_rev	lovebear_marg	birthdaybear_rev	birthdaybear_marg	minibear_rev	minibear_marg	total_revenue	total_margin
2013	9	26844.63	16378.50	5879.02	3675.00	NULL	NULL	NULL	NULL	32723.65	20053.50
2013	10	30143.97	18391.50	8098.65	5062.50	NULL	NULL	NULL	NULL	38242.62	23454.00
2013	11	36192.76	22082.00	10438.26	6525.00	NULL	NULL	NULL	NULL	46631.02	28607.00
2013	12	40891.82	24949.00	10978.17	6862.50	6392.61	4378.50	NULL	NULL	58262.60	36190.00
2014	1	36392.72	22204.00	10978.17	6862.50	9198.00	6300.00	NULL	NULL	56568.89	35366.50
2014	2	29194.16	17812.00	21056.49	13162.50	9703.89	6646.50	6057.98	4141.00	66012.52	41762.00
2014	3	39242.15	23942.50	11578.07	7237.50	11221.56	7686.00	6147.95	4202.50	68189.73	43068.50
2014	4	45840.83	27968.50	12837.86	8025.00	12279.33	8410.50	7767.41	5309.50	78725.43	49713.50
2014	5	51489.70	31415.00	14757.54	9225.00	13751.01	9418.50	8937.02	6109.00	88935.27	56167.50
2014	6	44641.07	27236.50	14697.55	9187.50	13245.12	9072.00	7467.51	5104.50	80051.25	50600.50
2014	7	48040.39	29310.50	14637.56	9150.00	12693.24	8694.00	7917.36	5412.00	83288.55	52566.50
2014	8	47890.42	29219.00	14217.63	8887.50	13521.06	9261.00	9086.97	6211.50	84716.08	53579.00
2014	9	52789.44	32208.00	15057.49	9412.50	14578.83	9985.50	9806.73	6703.50	92232.49	58309.50
2014	10	58638.27	35776.50	17037.16	10650.00	16924.32	11592.00	11306.23	7728.50	103905.98	65747.00
2014	11	72535.49	44255.50	22616.23	14137.50	19545.75	13387.50	13465.51	9204.50	128162.98	80985.00
2014	12	79184.16	48312.00	23216.13	14512.50	24788.61	16978.50	17634.12	12054.00	144823.02	91857.00
2015	1	69586.08	42456.00	23636.06	14775.00	20695.50	14175.00	18293.90	12505.00	132211.54	83911.00
2015	2	55638.87	33946.50	38633.56	24150.00	18625.95	12757.50	16314.56	11152.00	129212.94	82006.00

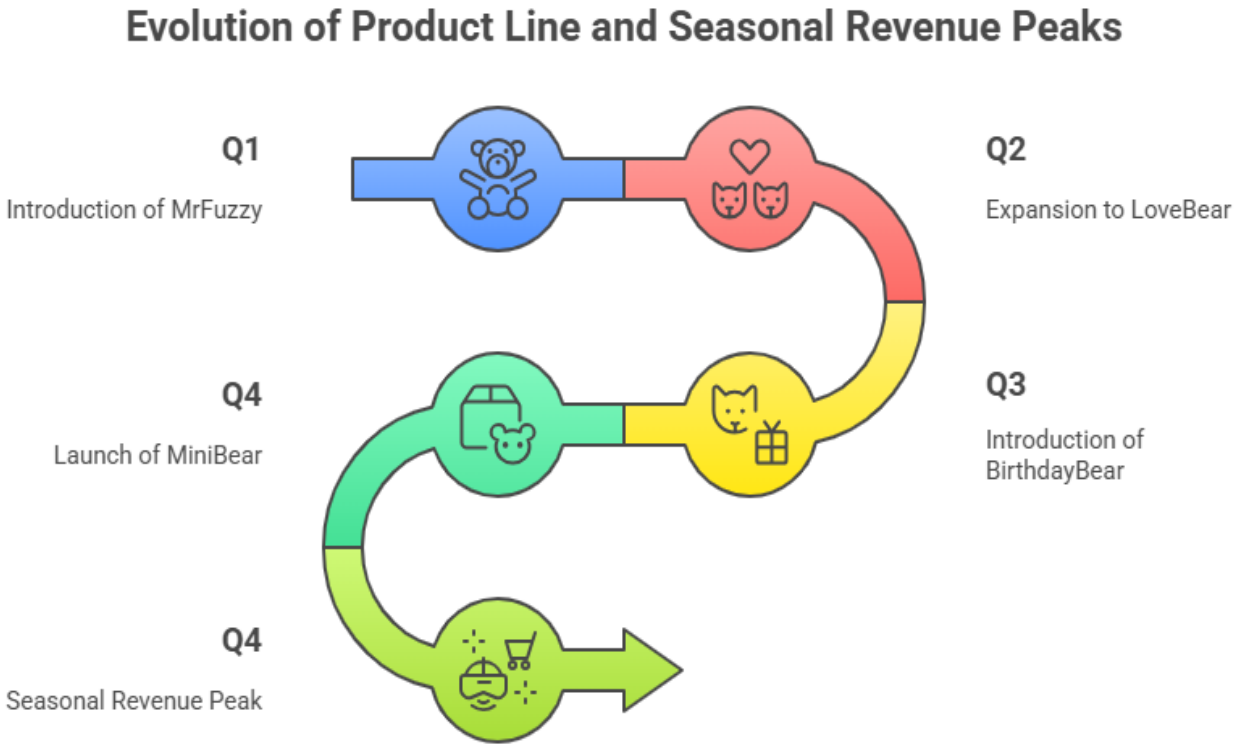
Insights:

- Product lineup expanded over time (MrFuzzy → LoveBear → BirthdayBear → MiniBear).
- Revenue and margin peaked seasonally (notably in Q4—likely due to holidays).
- Newer products like MiniBear show high revenue/margin in late months.
- Data shows Seasonality - November-December spikes

Recommendations:

✅ Plan product launches around peak months (Oct–Dec).

✅ Use bundling and upsell tactics for high-margin products for Q4



Q6. 🔗 Pull monthly sessions to the products page, and show how the % of those sessions clicking through another page has changed over time, along with a view of how conversion from products to placing an order has improved.

```sql
-- first, identifying all the views of the /products page
CREATE TEMPORARY TABLE products_pageviews
SELECT
    website_session_id,
    website_pageview_id,
    created_at AS saw_product_page_at

FROM website_pageviews
WHERE pageview_url = '/products';


SELECT
    YEAR(saw_product_page_at) AS yr,
    MONTH(saw_product_page_at) AS mo,
    COUNT(DISTINCT products_pageviews.website_session_id) AS sessions_to_product_page,
    COUNT(DISTINCT website_pageviews.website_session_id) AS clicked_to_next_page,
   round( COUNT(DISTINCT website_pageviews.website_session_id)*100/COUNT(DISTINCT
products_pageviews.website_session_id),2) AS clickthrough_rt,
    COUNT(DISTINCT orders.order_id) AS orders,
    round(COUNT(DISTINCT orders.order_id)*100/COUNT(DISTINCT products_pageviews.website_session_id),2) AS
products_to_order_rt
FROM products_pageviews
    LEFT JOIN website_pageviews
        ON website_pageviews.website_session_id = products_pageviews.website_session_id -- same session
        AND website_pageviews.website_pageview_id > products_pageviews.website_pageview_id -- they had
another page AFTER
    LEFT JOIN orders
        ON orders.website_session_id = products_pageviews.website_session_id
GROUP BY 1,2;
```

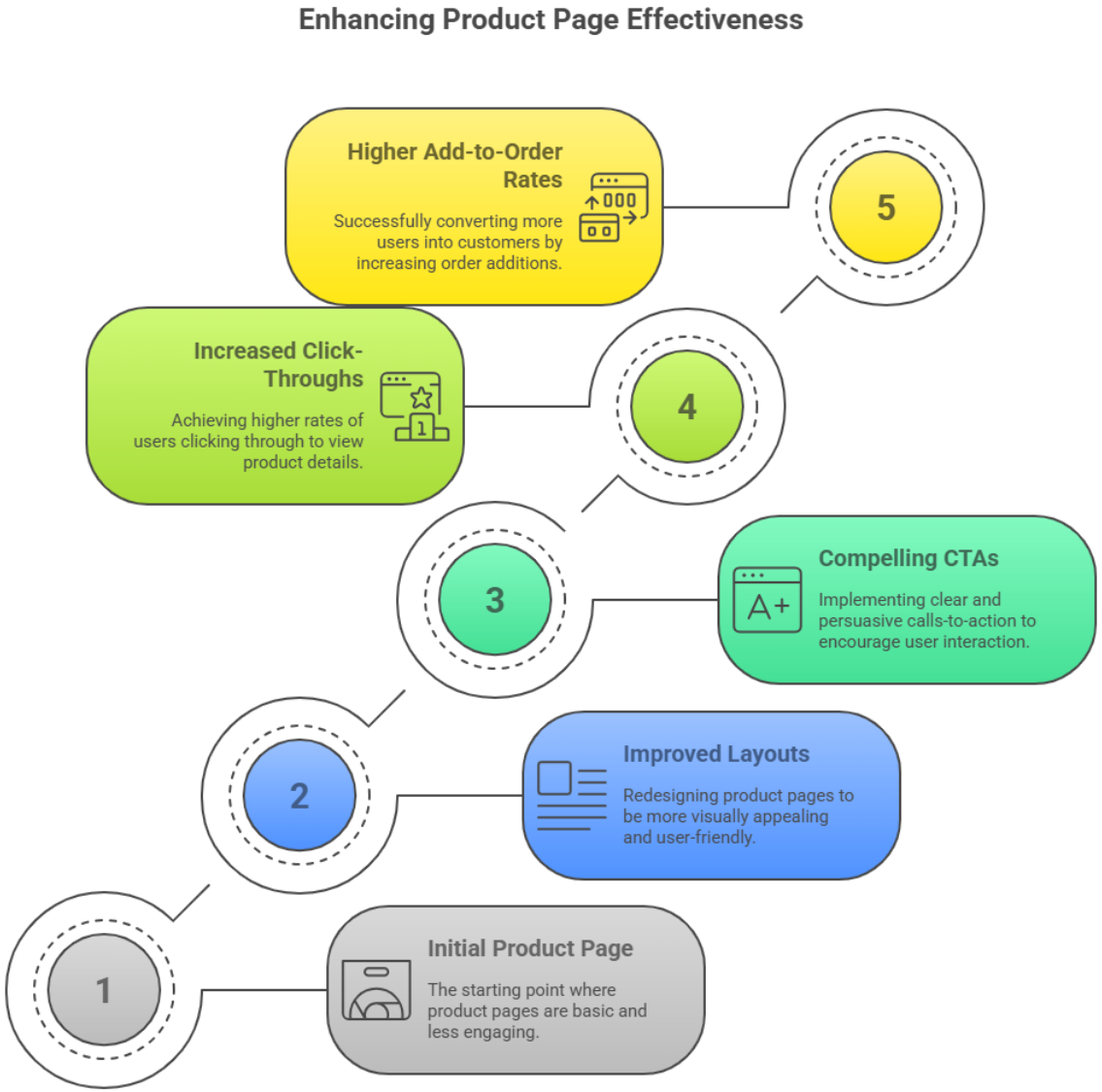
yr	mo	sessions_to_product_page	clicked_to_next_page	clickthrough_rt	orders	products_to_order_rt
2012	3	743	530	71.33	60	8.08
2012	4	1447	1029	71.11	99	6.84
2012	5	1584	1135	71.65	108	6.82
2012	6	1752	1247	71.18	140	7.99
2012	7	2018	1438	71.26	169	8.37
2012	8	3012	2198	72.97	228	7.57
2012	9	3126	2258	72.23	287	9.18
2012	10	4030	2948	73.15	371	9.21
2012	11	6743	4849	71.91	618	9.17
2012	12	5013	3620	72.21	506	10.09
2013	1	3380	2595	76.78	391	11.57
2013	2	3685	2803	76.07	497	13.49
2013	3	3371	2576	76.42	385	11.42
2013	4	4362	3356	76.94	553	12.68
2013	5	4684	3609	77.05	571	12.19
2013	6	4600	3536	76.87	594	12.91
2013	7	5020	3890	77.49	603	12.01
2013	8	5226	3951	75.60	608	11.63
2013	10	6038	4564	75.59	708	11.73
2013	11	7886	5900	74.82	861	10.92
2013	12	8840	7026	79.48	1047	11.84
2014	1	7790	6387	81.99	983	12.62
2014	2	7960	6485	81.47	1021	12.83
2014	3	8110	6669	82.23	1065	13.13
2014	4	9744	7958	81.67	1241	12.74
2014	5	10261	8465	82.50	1368	13.33
2014	6	10011	8260	82.51	1239	12.38
2014	7	10837	8958	82.66	1287	11.88
2014	8	10768	8980	83.40	1324	12.30
2014	9	11128	9156	82.28	1424	12.80
2014	10	12335	10235	82.98	1609	13.04
2014	11	14476	12020	83.03	1985	13.71
2014	12	17240	14609	84.74	2314	13.42
2015	1	15217	12992	85.38	2099	13.79
2015	2	14373	12187	84.79	2067	14.38
2015	3	9022	7723	85.60	1254	13.90

Insights:

Our product pages are becoming more effective. We've seen a steady improvement in both the rate at which customers click through from product pages and the rate at which they add items to their orders which suggests better product layouts or more compelling CTAs.

Recommendations:

✅ **Simplify Checkout**:
Reduce form fields, add guest checkout.
✅ **Add Trust Signals**:
Free shipping badges, secure payment icons.
✅ **Retarget Abandoned Carts**:
Send discount emails to users who didn't complete purchase.
Recommendation: A/B test different product page layouts, images, and calls-to-action to further improve these metrics.



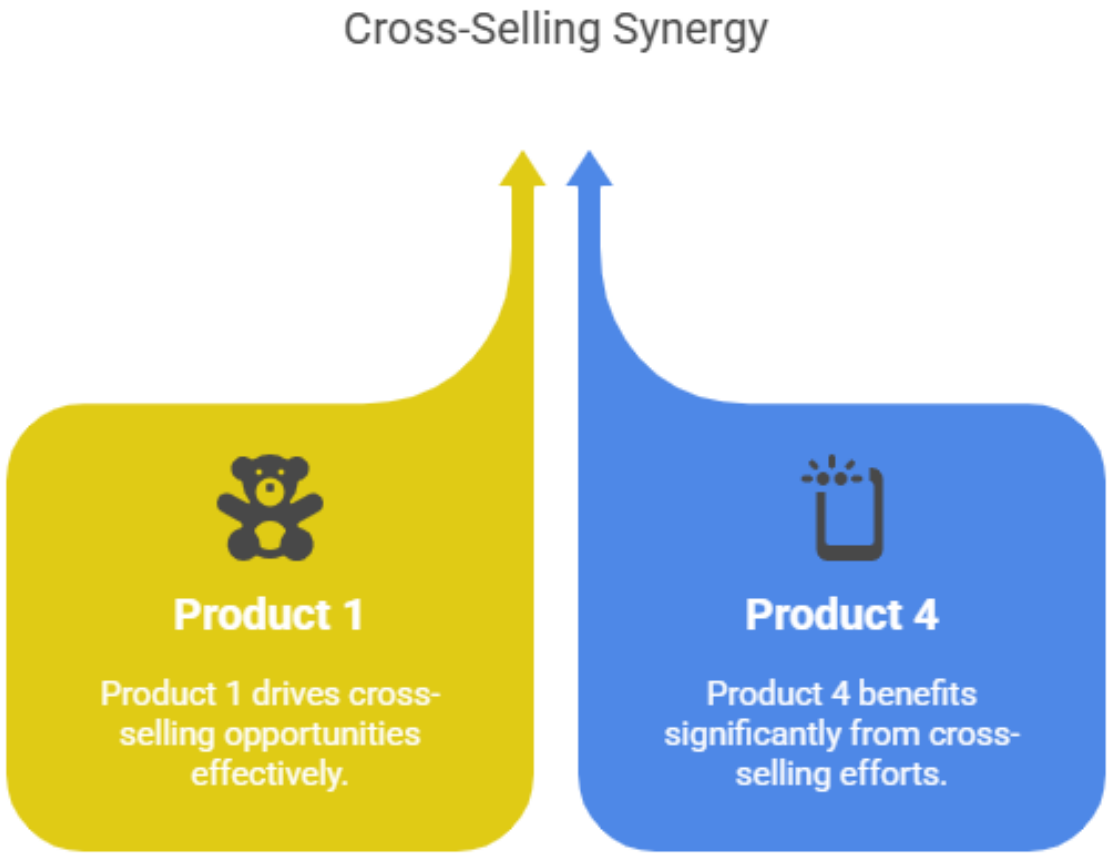
Q7. 🔄 Since Product 4 became a primary item on Dec 5,2014, how well do products cross-sell with one another since then?

```sql
CREATE TEMPORARY TABLE primary_products
SELECT
    order_id,
    primary_product_id,
    created_at AS ordered_at
FROM orders
WHERE created_at > '2014-12-05'; -- when the 4th product was added (says so in question)


SELECT
    primary_products.*,
    order_items.product_id AS cross_sell_product_id
FROM primary_products
    LEFT JOIN order_items
        ON order_items.order_id = primary_products.order_id
        AND order_items.is_primary_item = 0; -- only bringing in cross-sells;



SELECT
    primary_product_id,
    COUNT(DISTINCT order_id) AS total_orders,
    COUNT(DISTINCT CASE WHEN cross_sell_product_id = 1 THEN order_id ELSE NULL END) AS _xsold_p1,
    COUNT(DISTINCT CASE WHEN cross_sell_product_id = 2 THEN order_id ELSE NULL END) AS _xsold_p2,
    COUNT(DISTINCT CASE WHEN cross_sell_product_id = 3 THEN order_id ELSE NULL END) AS _xsold_p3,
    COUNT(DISTINCT CASE WHEN cross_sell_product_id = 4 THEN order_id ELSE NULL END) AS _xsold_p4,
    round(COUNT(DISTINCT CASE WHEN cross_sell_product_id = 1 THEN order_id ELSE NULL
END)*100/COUNT(DISTINCT order_id),2) AS p1_xsell_rt,
    round(COUNT(DISTINCT CASE WHEN cross_sell_product_id = 2 THEN order_id ELSE NULL
END)*100/COUNT(DISTINCT order_id),2) AS p2_xsell_rt,
    round(COUNT(DISTINCT CASE WHEN cross_sell_product_id = 3 THEN order_id ELSE NULL
END)*100/COUNT(DISTINCT order_id),2) AS p3_xsell_rt,
    round(COUNT(DISTINCT CASE WHEN cross_sell_product_id = 4 THEN order_id ELSE NULL
END)*100/COUNT(DISTINCT order_id),2) AS p4_xsell_rt
FROM
(
SELECT
    primary_products.*,
    order_items.product_id AS cross_sell_product_id
FROM primary_products
    LEFT JOIN order_items
        ON order_items.order_id = primary_products.order_id
        AND order_items.is_primary_item = 0 -- only bringing in cross-sells
) AS primary_w_cross_sell
GROUP BY 1;
```

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	0.00	5.33	12.38	20.89
2	1277	25	0	40	260	1.96	0.00	3.13	20.36
3	929	84	40	0	208	9.04	4.31	0.00	22.39
4	581	16	9	22	0	2.75	1.55	3.79	0.00

Insights:
Product 1 (MrFuzzy) has high cross-sell power (especially to Product 4).

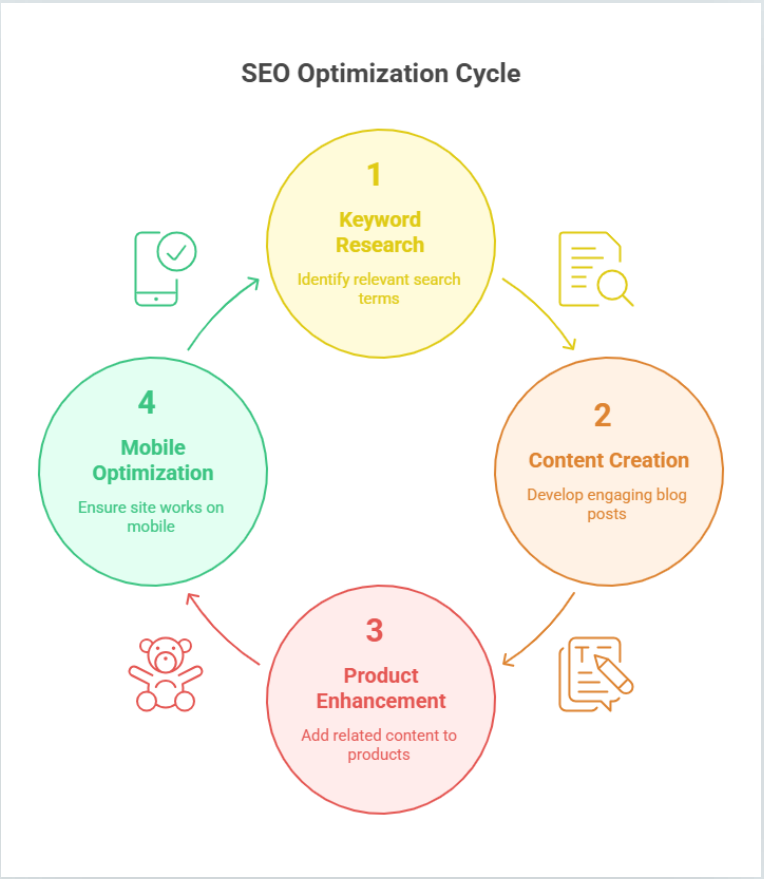
Recommendations:
Use Product 1 as a lead product in bundles or as a front-end hook.
✅ **Discount Cross-Sells**: Promote complementary pairing (e.g., MrFuzzy + MiniBear) with discounts. Add Mini Bear for 10% off entire order.
✅ **Promote Top Bundles**:
“Frequently Bought Together” prompts on product pages
✅ **Test New Pairings**:
Experiment with Product 2 + Product 4 promotions.



Q8. 🚀 Based on all analysis, what are the top opportunities and recommendations to drive future growth?

✅ Action Plan to Boost Ecommerce Growth (Recommendations)

```
1. 🚀 Double Down on SEO (Get More Free Traffic from Google)

🔍 Find keywords people actually search for (e.g., “soft teddy bear for birthday”)

✍️ Write helpful blog posts (e.g., gift guides, cleaning tips for stuffed animals)
🧸 Add related content around your products (e.g., teddy bear care, display ideas)

🛍️ Use keywords in product titles and descriptions

📱 Make sure your site works well on mobile
```



✅ Action Plan to Boost Ecommerce Growth (Recommendations)

2. 🛒 Optimize Conversion (Make It Easier to Buy)

📄 Simplify checkout (fewer steps, fewer forms)

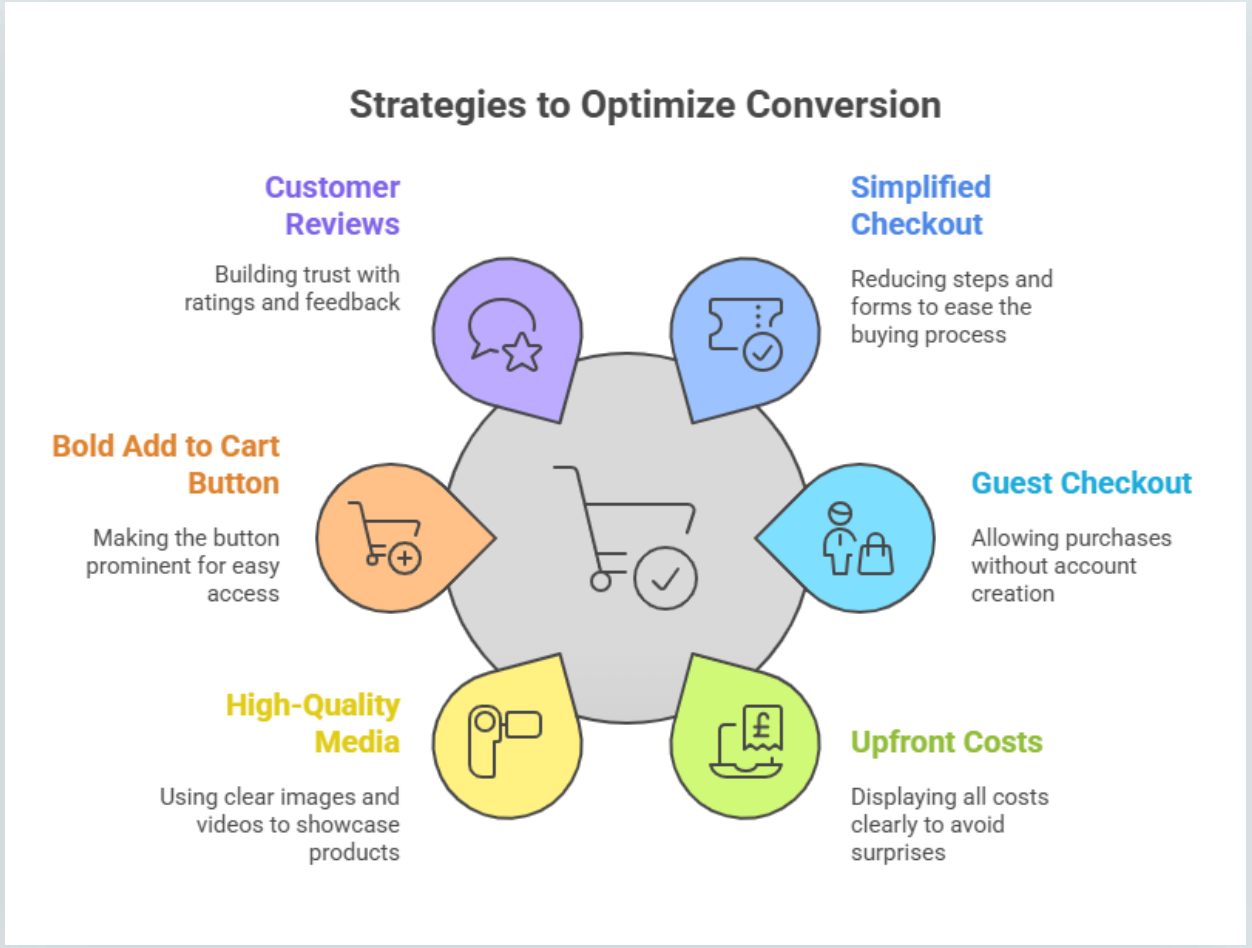
🕵️ Allow guest checkout (no forced account creation)

💸 Show all costs upfront (no surprise fees)

📸 Use high-quality product images and videos

🔘 Make the “Add to Cart” button big and bold

🌟 Add customer reviews and ratings for trust



✅ Action Plan to Boost Ecommerce Growth (Recommendations)

3. 💰 Increase Order Value with Cross-Selling (Encourage Bigger Carts)

🤝 Directly underneath the product description add a section that says "Frequently Bought Together" and shows a pre-ticked bundle at a slight discount

🎁 Bundle related items (e.g., teddy bear + gift box + card)

💡 Use pop-ups in the cart to suggest add-ons

🚚 Offer tiered shipping (e.g., “You’re ₹200 away from free shipping!”)

Instead of just "Free shipping on orders over $50," add a dynamic message in the cart like,
"You're only $12 away from free shipping!" This encourages people to add a small item to their cart to meet the threshold.

⚠️ Show low stock alerts (e.g., “Only 3 left!”)

Thank you