



# **ADVANCE SQL ANALYSIS OF CHANNEL PORTFOLIO AND BUSINESS PATTERNS**

**[WWW.LINKEDIN.COM/IN/SURAVSINGHHH/](https://www.linkedin.com/in/sauravsinghhh/)**

# **OBJECTIVE:**

## **CHANNEL PORTFOLIOS AND BUSINESS PATTERNS ANALYSIS**



**@SAURAVSINGHHHH**

# UNDERSTAND MAVEN FUZZY FACTORY DATABASE

website_sessions	website_pageviews	orders	order_items
<ul style="list-style-type: none"><li>website_session_id BIGINT</li><li>created_at DATETIME</li><li>user_id BIGINT</li><li>is_repeat_session BINARY</li><li>utm_source VARCHAR(45)</li><li>utm_campaign VARCHAR(45)</li><li>utm_content VARCHAR(45)</li><li>device_type VARCHAR(45)</li><li>http_referer VARCHAR(45)</li></ul>	<ul style="list-style-type: none"><li>website_pageview_id BIGINT</li><li>created_at DATETIME</li><li>website_session_id BIGINT</li><li>pageview_url VARCHAR(45)</li></ul> <p>Indexes</p>	<ul style="list-style-type: none"><li>order_id BIGINT</li><li>created_at DATETIME</li><li>website_session_id BIGINT</li><li>user_id BIGINT</li><li>primary_product_id INT</li><li>items_purchased INT</li><li>price_usd DECIMAL(6,2)</li><li>cogs_usd DECIMAL(6,2)</li></ul> <p>Indexes</p>	<ul style="list-style-type: none"><li>order_item_id BIGINT</li><li>created_at DATETIME</li><li>order_id BIGINT</li><li>product_id INT</li><li>is_primary_item BINARY</li><li>price_usd DECIMAL(6,2)</li><li>cogs_usd DECIMAL(6,2)</li></ul> <p>Indexes</p>
<p>Indexes</p>			

In this project, we will use  
these tables



@SAURAVSINGHHHH

# TOPICS COVERED IN THIS PROJECT

Sr no.	Used	Sr no.	Used
1	<i>Analyzing Channel Portfolios</i>	5	<i>Analyzing Product level website pathing</i>
2	<i>Analyzing direct, brand driven traffic</i>	6	<i>Cross-selling and Product Portfolio Analysis</i>
3	<i>Analyzing Business Patterns and Seasonality</i>	7	<i>User Level Analysis</i>
4	<i>Product level Analysis</i>	8	<i>Analyzing Repeat Visit &amp; Purchase Behaviour</i>

IN THIS ANALYSIS, I HAVE ANSWERED 7 QUESTIONS BASED ON THE INSIGHTS AND PATTERNS I HAVE ANALYZED



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[1/7]



First, I'd like to show our volume growth. Can you pull overall session and order volume, trended by quarter for the life of the business? Since the most recent quarter is incomplete, you can decide how to handle it.



@SAURAVSINGHHHH



```
SELECT  
    YEAR(website_sessions.created_at) AS yr,  
    QUARTER(website_sessions.created_at) AS qtr,  
    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,  
    COUNT(DISTINCT orders.order_id) AS orders  
FROM website_sessions  
    LEFT JOIN orders  
        ON website_sessions.website_session_id = orders.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

- Now as we are at the end of 3 years in business. we can see dramatic growth.
- We can see 60 orders from 1st Quarter. And in 2015 it has been increased 100 times more orders.
- There is similar growth in other Quarters too which is really impressive.



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[2/7]



Next, let's showcase all of our efficiency improvements. I would love to show quarterly figures since we launched, for session-to-order conversion rate, revenue per order, and revenue per session.



@SAURAVSINGHHHH



SELECT

```

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

```

yr	qtr	session_to_order_conv_rate	revenue_per_order	revenue_per_session
2012	1	0.0319	49.990000	1.596275
2012	2	0.0304	49.990000	1.517233
2012	3	0.0405	49.990000	2.024222
2012	4	0.0463	49.990000	2.316217
2013	1	0.0642	52.142396	3.346809
2013	2	0.0694	51.538312	3.578211
2013	3	0.0665	51.734533	3.441114
2013	4	0.0645	54.715688	3.530741
2014	1	0.0656	62.160684	4.078136
2014	2	0.0724	64.374207	4.662462
2014	3	0.0706	64.494949	4.554298
2014	4	0.0774	63.793497	4.934885
2015	1	0.0844	62.799917	5.301965

- Session growth Is a great story for this business as conversion rate of session to order grown from 3% to 8%.
- And Revenue per order increased from 49 dollars to above 60 dollars and in this the company started with only one product initially and they done some cross-selling and after optimization Revenue increased.
- Also Revenue per session grown from 1 dollars and it goes all the way up to 5 dollars.



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[3 / 7]



I'd like to show how we've grown specific channels. Could you pull a quarterly view of orders from Gsearch nonbrand, Bsearch nonbrand, brand search overall, organic search, and direct type-in?



@SAURAVSINGHHHH



```
SELECT
    YEAR(website_sessions.created_at) AS yr,
    QUARTER(website_sessions.created_at) AS qtr,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN orders.order_id ELSE NULL END)
    AS gsearch_nonbrand_orders,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN orders.order_id ELSE NULL END)
    AS bsearch_nonbrand_orders,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN orders.order_id ELSE NULL END) AS brand_search_orders,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN orders.order_id ELSE NULL END)
    AS organic_search_orders,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN orders.order_id ELSE NULL END)
    AS direct_type_in_orders
FROM website_sessions
LEFT JOIN orders
    ON website_sessions.website_session_id = orders.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

- No Surprise Here as we have seen tremendous growth over sessions previously.
- So it was coming from all of these channels.



@SAURAVSINGHHHH

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

- I think most of Investors are particularly excited about brand search, organic search and direct type in orders are picking up.
- Let focus on 2012 2nd quarter gsearch which high at that time and others channels (brand, organic, direct type in) orders were too less about 6:1 ratio.
- And in 2015 1st Quarter we can see gsearch and others ratios lies around 2:1 .
- Now it is less dependent on paid campaign.



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[4 / 7]



Next, let's show the overall session-to-order conversion rate trends for those same channels, by quarter. Please also make a note of any periods where we made major improvements or optimizations.



@SAURAVSINGHHHH



SELECT

```
YEAR(website_sessions.created_at) AS yr,  
QUARTER(website_sessions.created_at) AS qtr,  
COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN orders.order_id ELSE NULL END)  
/COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN  
website_sessions.website_session_id ELSE NULL END) AS gsearch_nonbrand_conv_rt,  
COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN orders.order_id ELSE NULL END)  
/COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN  
website_sessions.website_session_id ELSE NULL END) AS bsearch_nonbrand_conv_rt,  
COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN orders.order_id ELSE NULL END)  
/COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN  
website_sessions.website_session_id ELSE NULL END) AS brand_search_conv_rt,  
COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN orders.order_id ELSE NULL END)  
/COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NOT NULL THEN  
website_sessions.website_session_id ELSE NULL END) AS organic_search_conv_rt,  
COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN orders.order_id ELSE NULL END)  
/COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL THEN  
website_sessions.website_session_id ELSE NULL END) AS direct_type_in_conv_rt  
FROM website_sessions  
LEFT JOIN orders ON website_sessions.website_session_id = orders.website_session_id  
GROUP BY 1,2  
ORDER BY 1,2;
```



**This is the Query where orders by website sessions done of each channel to find conversion rate**



@SAURAVSINGHHHH

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	0.0324	NULL	0.0000	0.0000	0.0000
2012	2	0.0284	NULL	0.0526	0.0359	0.0536
2012	3	0.0384	0.0408	0.0602	0.0498	0.0443
2012	4	0.0436	0.0497	0.0531	0.0539	0.0537
2013	1	0.0612	0.0693	0.0703	0.0753	0.0614
2013	2	0.0685	0.0690	0.0679	0.0760	0.0735
2013	3	0.0639	0.0697	0.0703	0.0734	0.0719
2013	4	0.0629	0.0601	0.0801	0.0694	0.0647
2014	1	0.0693	0.0704	0.0839	0.0756	0.0765
2014	2	0.0702	0.0695	0.0804	0.0797	0.0738
2014	3	0.0703	0.0698	0.0756	0.0733	0.0702
2014	4	0.0782	0.0841	0.0812	0.0784	0.0748
2015	1	0.0861	0.0850	0.0852	0.0821	0.0775

- We can see gsearch conversion rate which increased from 3% to 8% in recent Quarters.
- Bsearch growth have same story too.
- Brand, organic and direct type in conversion rate all of them have seen substantial improvements from where they were initially to where they are now.
- It shows company is really working in growth rate to improve business.



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[5/7]



We've come a long way since the days of selling a single product. Let's pull monthly trending for revenue and margin by product, along with total sales and revenue. Note anything you notice about seasonality.



@SAURAVSINGHHHH



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

This Query finds  
the revenue of  
each toy with  
margin



@SAURAVSINGHHHH

yr	mo	mrfuzzy_rev	mrfuzzy_marg	lovebear_rev	lovebear_marg	birthdaybear_rev	birthdaybear_marg	minibear_rev	minibear_marg	total_revenue	total_margin
2013	6	25144.97	15341.50	5399.10	3375.00	MULL	MULL	MULL	MULL	30544.07	18716.50
2013	7	25444.91	15524.50	5699.05	3562.50	MULL	MULL	MULL	MULL	31143.96	19087.00
2013	8	25494.90	15555.00	5879.02	3675.00	MULL	MULL	MULL	MULL	31373.92	19230.00
2013	9	26844.63	16378.50	5879.02	3675.00	MULL	MULL	MULL	MULL	32723.65	20053.50
2013	10	30143.97	18391.50	8098.65	5062.50	MULL	MULL	MULL	MULL	38242.62	23454.00
2013	11	36192.76	22082.00	10438.26	6525.00	MULL	MULL	MULL	MULL	46631.02	28607.00
2013	12	40891.82	24949.00	10978.17	6862.50	6392.61	4378.50	MULL	MULL	58262.60	36190.00
2014	1	36392.72	22204.00	10978.17	6862.50	9198.00	6300.00	MULL	MULL	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
2015	3	43191.36	26352.00	13377.77	8362.50	12095.37	8284.50	10286.57	7031.50	78951.07	50030.50

- We can see mrfuzzy revenue was approx. 3000 dollar initially and it was increased to 43000 dollar approx. in 2015 Quarter.**
- Focusing on 2014 Nov and Dec month it has highest sales.**
- Love Bear Toy sales was highest in the month of February as this bear is targeted to couples for giving this toys as a gift.**
- Rest products shows major sales in the end of years.**



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[6/7]



Let's dive deeper into the impact of introducing new products. Please 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.



@SAURAVSINGHHHH



```
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 * FROM products_pageviews;
```

website_session_id	website_pageview_id	saw_product_page_at
6	8	2012-03-19 09:10:08
15	22	2012-03-19 10:01:46
16	24	2012-03-19 10:07:07
18	28	2012-03-19 10:18:58
20	32	2012-03-19 10:28:09
27	46	2012-03-19 11:28:43
30	50	2012-03-19 11:47:49
33	53	2012-03-19 11:59:59
34	57	2012-03-19 12:03:01

## PART:[1/2]



We have whole lots of session and website pageviews and when they saw that product page.



@SAURAVSINGHHHH



```
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,  
    COUNT(DISTINCT website_pageviews.website_session_id)  
        /COUNT(DISTINCT products_pageviews.website_session_id) AS clickthrough_rt,  
    COUNT(DISTINCT orders.order_id) AS orders,  
    COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT products_pageviews.website_session_id)  
        AS products_to_order_rt  
FROM products_pageviews  
    LEFT JOIN website_pageviews  
        ON website_pageviews.website_session_id = products_pageviews.website_session_id  
        AND website_pageviews.website_pageview_id > products_pageviews.website_pageview_id  
    LEFT JOIN orders  
        ON orders.website_session_id = products_pageviews.website_session_id  
GROUP BY 1,2;
```

PART:[2/2]

In this Query we will be focusing on  
clickthrough rate and product to order  
conversion rate



@SAURAVSINGHHHH

yr	mo	sessions_to_product_page	clicked_to_next_page	clickthrough_rt	orders	products_to_order_rt
2012	3	743	530	0.7133	60	0.0808
2012	4	1447	1029	0.7111	99	0.0684
2012	5	1584	1135	0.7165	108	0.0682
2012	6	1752	1247	0.7118	140	0.0799
2012	7	2018	1438	0.7126	169	0.0837
2012	8	3012	2198	0.7297	228	0.0757
2012	9	3126	2258	0.7223	287	0.0918
2012	10	4030	2948	0.7315	371	0.0921
2012	11	6743	4849	0.7191	618	0.0917
2012	12	5013	3620	0.7221	506	0.1009
2013	1	3380	2595	0.7678	391	0.1157
2013	2	3685	2803	0.7607	497	0.1349
2013	3	3371	2576	0.7642	385	0.1142
2013	4	4362	3356	0.7694	553	0.1268
2013	5	4684	3609	0.7705	571	0.1219
2013	6	4600	3536	0.7687	594	0.1291
2013	7	5020	3890	0.7749	603	0.1201
2013	8	5226	3951	0.7560	608	0.1163
2013	9	5399	4072	0.7542	629	0.1165
2013	10	6038	4564	0.7559	708	0.1173
2013	11	7886	5900	0.7482	861	0.1092
2013	12	8840	7026	0.7948	1047	0.1184

## PART:[2/2]

- So what we see here is year, month, session to product page, clicked to next page, clickthrough rate, orders, product to order rate.
- Main thing to observe here is clickthrough rate which was initially 71% which was increased to 85% in most recent months.
- Similarly in people seeing products and ordering it rate is increased from 8% to 14% over a year.



@SAURAVSINGHHHH

# QUESTIONS TO BE ANSWERED IN THIS ANALYSIS

[7/7]



We made our 4th product available as a primary product on December 05, 2014 (it was previously only a cross-sell item). Could you please pull sales data since then, and show how well each product cross-sells from one another?



@SAURAVSINGHHHH

```
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 * FROM primary_products;
```

order_id	primary_product_id	ordered_at
25087	1	2014-12-05 12:15:40
25088	1	2014-12-05 12:19:41
25089	1	2014-12-05 12:20:02
25090	1	2014-12-05 13:03:46
25091	2	2014-12-05 13:15:27
25092	3	2014-12-05 13:26:00
25093	2	2014-12-05 13:53:02
25094	1	2014-12-05 14:02:25
25095	1	2014-12-05 14:48:05
25096	1	2014-12-05 15:02:56
25097	1	2014-12-05 15:06:24
25098	3	2014-12-05 15:14:15
25099	1	2014-12-05 15:17:39

Here I have extracted primary product id and order timing.



## PART:[2/3]

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;
```

order_id	primary_product_id	ordered_at	cross_sell_product_id
25060	1	2014-12-05 01:10:46	4
25061	1	2014-12-05 01:18:23	4
25062	1	2014-12-05 02:31:10	2
25063	3	2014-12-05 02:50:46	NULL
25064	1	2014-12-05 02:54:53	3
25065	1	2014-12-05 04:04:00	3
25066	1	2014-12-05 04:17:33	4
25067	1	2014-12-05 05:29:39	3
25068	2	2014-12-05 05:51:20	NULL
25069	2	2014-12-05 06:42:42	NULL
25070	1	2014-12-05 07:00:43	NULL
25071	1	2014-12-05 07:48:08	NULL
25072	1	2014-12-05 07:50:12	NULL
25073	1	2014-12-05 07:57:10	NULL
25074	1	2014-12-05 08:20:22	NULL
25075	1	2014-12-05 09:07:18	4

- And now what we see in this query that we have a order id, primary product id and which products was cross sold.
- We can observe product 4 and 2 are cross sold. Null value shows the product wasn't cross sold.



@SAURAVSINGHHHH



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,  
COUNT(DISTINCT CASE WHEN cross_sell_product_id = 1 THEN order_id ELSE NULL END)  
/COUNT(DISTINCT order_id) AS p1_xsell_rt,  
COUNT(DISTINCT CASE WHEN cross_sell_product_id = 2 THEN order_id ELSE NULL END)  
/COUNT(DISTINCT order_id) AS p2_xsell_rt,  
COUNT(DISTINCT CASE WHEN cross_sell_product_id = 3 THEN order_id ELSE NULL END)  
/COUNT(DISTINCT order_id) AS p3_xsell_rt,  
COUNT(DISTINCT CASE WHEN cross_sell_product_id = 4 THEN order_id ELSE NULL END)  
/COUNT(DISTINCT order_id) 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;
```

## PART:[3/3]

This Query finds the number of cross selling of each product. Also it Finds the Rate at which cross selling done of products



@SAURAVSINGHHHH

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

- We got here the primary product id which I have grouped by product 1,2,3,4.
- Then we have total orders of each products.
- We can observe here that product 1 was never cross sold when product 1 was the primary product.
- After that Product 1 was cross sold with product 2 ,3,4 in some amount.
- Seeing Cross selling of Product 4 which was sold in high volumes.
- One of the reason was Product 4 have low price so its cross selling increased.
- Then the same numbers were translated into rates as to the percentage of the total orders.
- Here Product 3 and 4 was cross sold really well.



PART:[3/3]



@SAURAVSINGHHHH

# THANK'S FOR REVIEWING

Connect with me on:



**LINKEDIN.COM/IN/SAURAVSINGHHHH**



**GITHUB.COM/SAURAV2021C**



**TWITTER.COM/SAURAVSINGHHHS**