My (Jamie Lyons) overview/report of an analysis for an online businesses fundraising round!

- **Situation At Hand : Help The Business Secure A Funding Round By Telling The Growth Story!**
- **Extract The Data And Tell The Story So We Can Help Advise The C-Suite How They Can Use The Data To Persuade Potential Investors**

Query 1 :

- Gather Session And Order Volume Trended By Quarter For The Life Of The Business
- Check The Conversion Rate, Revenue Per Order And Revenue Per Session

```
SELECT
    YEAR(ws.created_at) AS yr,
    QUARTER(ws.created_at) AS q,
    COUNT(DISTINCT ws.website_session_id) AS num_sessions,
    COUNT(DISTINCT o.order_id) AS num_orders,
    ROUND(COUNT(DISTINCT o.order_id) / COUNT(DISTINCT ws.website_session_id),3) AS
cvr_rt,
    ROUND(SUM(price_usd) / COUNT(DISTINCT o.order_id),2) AS rev_per_order,
    ROUND(SUM(price_usd) / COUNT(DISTINCT ws.website_session_id),2) AS
rev_per_session
FROM
    website_sessions ws
-- bringing is website_sessions for access to session level data
    LEFT JOIN orders o
          USING(website_session_id)
-- joining to the orders table so we can create the conversion rates and bring in revenue figures
WHERE
     ws.created_at <= '2014-12-31'
GROUP BY
-- grouping by year and quarter to create the trending analysis possible
     1,2
ORDER BY
     1,2  ;
-- ordering by year and quarter to ensure it is in chronological order ;
```

yr	q	num_sessions	num_orders	cvr_rt	rev_per_order	rev_per_session
2012	1	1879	60	0.032	49.99	1.60
2012	2	11433	347	0.030	49.99	1.52
2012	3	16892	684	0.040	49.99	2.02
2012	4	32266	1495	0.046	49.99	2.32
2013	1	19833	1273	0.064	52.14	3.35
2013	2	24745	1718	0.069	51.54	3.58
2013	3	27663	1840	0.067	51.73	3.44
2013	4	40540	2616	0.065	54.72	3.53
2014	1	46779	3069	0.066	62.16	4.08
2014	2	53129	3848	0.072	64.37	4.66
2014	3	57141	4035	0.071	64.49	4.55
2014	4	75434	5836	0.077	63.80	4.94

ANALYSIS OF QUERY 1:

- Ordering by year and quarter to make align with the desired trend pattern
- You can see a clear trend here, record quarters consistently for 3 years.
- Not only that, during that time the conversion rate from sessions to orders has increased around 2.5x from a low of 3% to 7.8%
- Orders has risen from 60 5420 in 3 years, almost 100x increase
- Revenue per session has increased from 1.6 to 5.3 (over 200%)
- Revenue per order has increase from 50 to 62, through increasing product variety with huge success

NOTE : the recent quarter is incomplete, I have decided to leave this out to maintain consistency in the data, alternatively we could have predicted the outcome.

Query 2:

- Gather Trended Views Of Orders That Came From Gsearch Nonbrand, Bsearch Nonbrand, Brand Search Overall, Organic Search And Direct Type In So We Can Assess How The Brand Is Gaining Strength And Traffic Is Coming To Our Website

```
SELECT
    YEAR(ws.created_at) AS yr,
    QUARTER(ws.created_at) AS q,
    COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign= 'nonbrand'
THEN o.order_id ELSE NULL END) AS gsearch_nonbrand,
    COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign= 'nonbrand'
THEN o.order_id ELSE NULL END) AS bsearch_nonbrand,
```

```
    COUNT(DISTINCT CASE WHEN http_referer IS NOT NULL AND utm_source IS NULL
THEN o.order_id ELSE NULL END) AS organic_search,
    COUNT(DISTINCT CASE WHEN http_referer IS NULL THEN o.order_id ELSE NULL END)
AS direct_type_in,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN o.order_id ELSE NULL
END) AS total_brand_search
-- creating the buckets for each traffic type, to assess and analyse their performance
FROM
     website_sessions ws
     LEFT JOIN orders o
          USING(website_session_id)
WHERE
     ws.created_at <= '2014-12-31'
GROUP BY
   YEAR(ws.created_at),
   QUARTER(ws.created_at)
ORDER BY
   YEAR(ws.created_at),
   QUARTER(ws.created_at)
;
```

yr	q	gsearch_nonbrand	bsearch_nonbrand	organic_search	direct_type_in	total_brand_search
2012	1	60	0	0	0	0
2012	2	291	0	15	21	20
2012	3	482	82	40	32	48
2012	4	913	311	94	89	88
2013	1	766	183	125	91	108
2013	2	1114	237	134	119	114
2013	3	1132	245	167	143	153
2013	4	1657	291	223	197	248
2014	1	1667	344	338	311	354
2014	2	2208	427	436	367	410
2014	3	2259	434	445	402	432
2014	4	3207	679	594	525	606

```
-- using the previous query to translate into what % of the orders are coming from each traffic
type

SELECT
   YEAR(ws.created_at) AS yr,
   QUARTER(ws.created_at) AS q,
    ROUND(COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign=
'nonbrand' THEN o.order_id ELSE NULL END) / COUNT(DISTINCT o.order_id),2) AS
gsearch_nonbrand_pct,
```

```
    ROUND(COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign=
'nonbrand' THEN o.order_id ELSE NULL END) / COUNT(DISTINCT o.order_id),2) AS
bsearch_nonbrand_pct,
    ROUND(COUNT(DISTINCT CASE WHEN http_referer IS NOT NULL AND utm_source IS
NULL THEN o.order_id ELSE NULL END) / COUNT(DISTINCT o.order_id),2) AS
organic_search_pct,
    ROUND(COUNT(DISTINCT CASE WHEN http_referer IS NULL THEN o.order_id ELSE
NULL END) / COUNT(DISTINCT o.order_id),2) AS direct_type_in_pct,
    ROUND(COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN o.order_id ELSE
NULL END) / COUNT(DISTINCT o.order_id),2) AS total_brand_search_pct
-- rounded to 2 decimals for easier readability on the end user
FROM
     website_sessions ws
     LEFT JOIN orders o
          USING(website_session_id)
WHERE
   ws.created_at <= '2014-12-31'
GROUP BY
   YEAR(ws.created_at),
   QUARTER(ws.created_at)
ORDER BY
   YEAR(ws.created_at),
   QUARTER(ws.created_at)
;
```

yr	q	gsearch_nonbrand_pct	bsearch_nonbrand_pct	organic_search_pct	direct_type_in_pct	total_brand_search_pct
2012	1	1.00	0.00	0.00	0.00	0.00
2012	2	0.84	0.00	0.04	0.06	0.06
2012	3	0.70	0.12	0.06	0.05	0.07
2012	4	0.61	0.21	0.06	0.06	0.06
2013	1	0.60	0.14	0.10	0.07	0.08
2013	2	0.65	0.14	0.08	0.07	0.07
2013	3	0.62	0.13	0.09	0.08	0.08
2013	4	0.63	0.11	0.09	0.08	0.09
2014	1	0.54	0.11	0.11	0.10	0.12
2014	2	0.57	0.11	0.11	0.10	0.11
2014	3	0.56	0.11	0.11	0.10	0.11
2014	4	0.55	0.12	0.10	0.09	0.10

ANALYSIS OF QUERY 2:

- Direct type in numbers has increase 25 x per quarter over the 3 years, organic search has increased by even more than that
- Paid channels are taking up less and less of the share of orders, which has better margain and less dependency on paid traffic, shows brand that our health is getting stronger and customers are coming through favorable channels such as organic or direct type in

Query 3 :

- Gathering Quarterly Session To Order From Each Gsearch Nonbrand, Bsearch Nonbrand, Brand Search Overall, Organic Search And Direct Type In?

```
SELECT
    YEAR(ws.created_at) AS yr,
    QUARTER(ws.created_at) AS q,
-- now creating the formula to calculate what percentage of the orders came from each type of traffic below
     ROUND(COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign= 'nonbrand' THEN o.order_id ELSE NULL END)
                    / COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND utm_campaign= 'nonbrand' THEN ws.website_session_id ELSE NULL END),3)  AS gsearch_nonbrand,
     ROUND(COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN o.order_id ELSE NULL END)
                    / COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND utm_campaign = 'nonbrand' THEN ws.website_session_id ELSE NULL END),3)  AS bsearch_nonbrand,
    ROUND(COUNT(DISTINCT CASE WHEN http_referer IS NOT NULL AND utm_source IS NULL THEN o.order_id ELSE NULL END)
                    / COUNT(DISTINCT CASE WHEN http_referer IS NOT NULL AND utm_source IS NULL THEN ws.website_session_id ELSE NULL END),3)  AS organic_search,
    ROUND(COUNT(DISTINCT CASE WHEN http_referer IS NULL THEN o.order_id ELSE NULL END)
                    / COUNT(DISTINCT CASE WHEN http_referer IS NULL THEN ws.website_session_id ELSE NULL END),3) AS direct_type_in,
     ROUND(COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN o.order_id ELSE NULL END)
                    / COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN ws.website_session_id ELSE NULL END),3)  AS total_brand_search
FROM
     website_sessions ws
          LEFT JOIN orders o
                    USING(website_session_id)
-- bringing in website sessions to create the groups of each traffic type, and orders to figure out which sessions resulted in completing orders
WHERE
     ws.created_at <= '2014-12-31'
-- limiting the date range to only completed quarters
GROUP BY
     1,2 ;
```

yr	q	gsearch_nonbrand	bsearch_nonbrand	organic_search	direct_type_in	total_brand_search
2012	1	0.032	NULL	0.000	0.000	0.000
2012	2	0.028	NULL	0.036	0.054	0.053
2012	3	0.038	0.041	0.050	0.044	0.060
2012	4	0.044	0.050	0.054	0.054	0.053
2013	1	0.061	0.069	0.075	0.061	0.070
2013	2	0.069	0.069	0.076	0.074	0.068
2013	3	0.064	0.070	0.073	0.072	0.070
2013	4	0.063	0.060	0.069	0.065	0.080
2014	1	0.069	0.070	0.076	0.077	0.084
2014	2	0.070	0.069	0.080	0.074	0.080
2014	3	0.070	0.070	0.073	0.070	0.076
2014	4	0.078	0.084	0.078	0.075	0.081

ANALYSIS OF QUERY 3:

- Across all of our channels our session to order conversion rate is steadily growing, an indication of our previous analysis and website improvements / purchase pathway optimisations. A clear indicator of that being in january 2013 improvements of our billing page causing a considerable uplift of around 2-3% to prior quarters

Query 4 :

- Diving Deeper Into Revenue By Products, Let's See What The Monthly Sales Trends Look Like And Potential High Seasons

```
SELECT
    YEAR(oi.created_at) AS yr,
    MONTH(oi.created_at) AS mo,
    SUM(CASE WHEN oi.product_id = '1' THEN oi.price_usd ELSE NULL END) AS p1_revenue,
    SUM(CASE WHEN oi.product_id = '1' THEN oi.price_usd - cogs_usd ELSE NULL END)  AS
p1_margain,
    SUM(CASE WHEN oi.product_id = '2' THEN oi.price_usd ELSE NULL END) AS p2_revenue,
    SUM(CASE WHEN oi.product_id = '2' THEN oi.price_usd - cogs_usd ELSE NULL END) AS
p2_margain,
    SUM(CASE WHEN oi.product_id = '3' THEN oi.price_usd ELSE NULL END) AS p3_revenue,
    SUM(CASE WHEN oi.product_id = '3' THEN oi.price_usd - cogs_usd ELSE NULL END) AS
p3_margain,
    SUM(CASE WHEN oi.product_id = '4' THEN oi.price_usd ELSE NULL END) AS p4_revenue,
    SUM(CASE WHEN oi.product_id = '4' THEN oi.price_usd - cogs_usd ELSE NULL END)AS
p4_margain,
-- using case to determine which values to sum for each product, to then assess revenue and
margin for each
    COUNT(oi.order_id) AS total_sales,
    SUM(oi.price_usd) AS revenue,
    SUM(oi.price_usd - cogs_usd) AS margain
```

```
-- some additional lines to give a higher level view and context of overall sales (not just product level)
FROM
    order_items oi
WHERE
    oi.created_at <= '2015-02-28'
GROUP BY
    1,2
;
```

yr	mo	p1_revenue	p1_margin	p2_revenue	p2_margin	p3_revenue	p3_margin	p4_revenue	p4_margin	total_sales	revenue	margain
2012	3	2999.40	1830.00	NULL	NULL	NULL	NULL	NULL	NULL	60	2999.40	1830.00
2012	4	4949.01	3019.50	NULL	NULL	NULL	NULL	NULL	NULL	99	4949.01	3019.50
2012	5	5398.92	3294.00	NULL	NULL	NULL	NULL	NULL	NULL	108	5398.92	3294.00
2012	6	6998.60	4270.00	NULL	NULL	NULL	NULL	NULL	NULL	140	6998.60	4270.00
2012	7	8448.31	5154.50	NULL	NULL	NULL	NULL	NULL	NULL	169	8448.31	5154.50
2012	8	11397.72	6954.00	NULL	NULL	NULL	NULL	NULL	NULL	228	11397.72	6954.00
2012	9	14347.13	8753.50	NULL	NULL	NULL	NULL	NULL	NULL	287	14347.13	8753.50
2012	10	18546.29	11315.50	NULL	NULL	NULL	NULL	NULL	NULL	371	18546.29	11315.50
2012	11	30893.82	18849.00	NULL	NULL	NULL	NULL	NULL	NULL	618	30893.82	18849.00
2012	12	25294.94	15433.00	NULL	NULL	NULL	NULL	NULL	NULL	506	25294.94	15433.00
2013	1	17146.57	10461.50	2819.53	1762.50	NULL	NULL	NULL	NULL	390	19966.10	12224.00
2013	2	16796.64	10248.00	9718.38	6075.00	NULL	NULL	NULL	NULL	498	26515.02	16323.00
2013	3	15996.80	9760.00	3899.35	2437.50	NULL	NULL	NULL	NULL	385	19896.15	12197.50
2013	4	22945.41	13999.50	5639.06	3525.00	NULL	NULL	NULL	NULL	553	28584.47	17524.50
2013	5	24445.11	14914.50	4919.18	3075.00	NULL	NULL	NULL	NULL	571	29364.29	17989.50
2013	6	25144.97	15341.50	5399.10	3375.00	NULL	NULL	NULL	NULL	593	30544.07	18716.50
2013	7	25444.91	15524.50	5699.05	3562.50	NULL	NULL	NULL	NULL	604	31143.96	19087.00
2013	8	25494.90	15555.00	5879.02	3675.00	NULL	NULL	NULL	NULL	608	31373.92	19230.00
2013	9	26844.63	16378.50	5879.02	3675.00	NULL	NULL	NULL	NULL	635	32723.65	20053.50
2013	10	30143.97	18391.50	8098.65	5062.50	NULL	NULL	NULL	NULL	738	38242.62	23454.00
2013	11	36192.76	22082.00	10438.26	6525.00	NULL	NULL	NULL	NULL	898	46631.02	28607.00
2013	12	40891.82	24949.00	10978.17	6862.50	6392.61	4378.50	NULL	NULL	1140	58262.60	36190.00
2014	1	36392.72	22204.00	10978.17	6862.50	9198.00	6300.00	NULL	NULL	1111	56568.89	35366.50
2014	2	29194.16	17812.00	21056.49	13162.50	9703.89	6646.50	6057.98	4141.00	1348	66012.52	41762.00
2014	3	39242.15	23942.50	11578.07	7237.50	11221.56	7686.00	6147.95	4202.50	1427	68189.73	43068.50
2014	4	45840.83	27968.50	12837.86	8025.00	12279.33	8410.50	7767.41	5309.50	1657	78725.43	49713.50
2014	5	51489.70	31415.00	14757.54	9225.00	13751.01	9418.50	8937.02	6109.00	1873	88935.27	56167.50
2014	6	44641.07	27236.50	14697.55	9187.50	13245.12	9072.00	7467.51	5104.50	1675	80051.25	50600.50
2014	7	48040.39	29310.50	14637.56	9150.00	12693.24	8694.00	7917.36	5412.00	1745	83288.55	52566.50
2014	8	47890.42	29219.00	14217.63	8887.50	13521.06	9261.00	9086.97	6211.50	1792	84716.08	53579.00
2014	9	52789.44	32208.00	15057.49	9412.50	14578.83	9985.50	9806.73	6703.50	1951	92232.49	58309.50
2014	10	58638.27	35776.50	17037.16	10650.00	16924.32	11592.00	11306.23	7728.50	2202	103905.98	65747.00
2014	11	72535.49	44255.50	22616.23	14137.50	19545.75	13387.50	13465.51	9204.50	2702	128162.98	80985.00
2014	12	79184.16	48312.00	23216.13	14512.50	24788.61	16978.50	17634.12	12054.00	3098	144823.02	91857.00
2015	1	69586.08	42456.00	23636.06	14775.00	20695.50	14175.00	18293.90	12505.00	2846	132211.54	83911.00
2015	2	54539.09	33275.50	38273.62	23925.00	18304.02	12537.00	15924.69	10885.50	2658	127041.42	80623.00

ANALYSIS OF QUERY 4:

- Every year, from start to finish we see considerable growth of around 200% increase in orders, typically with a large surge around the christmas period and later around valentines once the love bear (product 2) was launched in 2013 Jan. This suggests that we are tapping into the lucrative gifts market (Christmas + Valentines)
- Overall monthly revenue is up 43x since launch

NOTE : again capping off at the final month as we don't have a full story of data yet

Query 5 :

- Check How Monthly Sessions On The Products Page Are Growing And What % Are Clicking Through To Another Page Look How Those Sessions Are Converting To Orders

```sql
CREATE TEMPORARY TABLE click_throughs
-- creating a temporary table to flag those website sessions that clicked through to the next page
SELECT
    prod_page.website_session_id,
    products_pageview_id,
    prod_created_at,
    MIN(pvs.website_pageview_id) AS clicked_through,
    MIN(pvs.created_at) AS click_through_created_at
FROM
        (
-- first of all creating a subquery in order to select the MIN page_id that is on the product page
  then we can see if any subsequent pages were clicked in the main query
                SELECT
                        website_session_id,
                        MIN(website_pageview_id) AS products_pageview_id,
                        MIN(created_at) AS prod_created_at
                FROM
                        website_pageviews
                WHERE
                        pageview_url = '/products'
                        AND created_at <= '2015-02-28'
-- limiting to products so that we can find out what pageview id each session was on the
products page and then grouped by session also
                GROUP BY
                        website_session_id
      ) AS prod_page
LEFT JOIN
        website_pageviews pvs
                ON pvs.website_session_id = prod_page.website_session_id
                AND pvs.website_pageview_id > prod_page.products_pageview_id
-- left joining to website pageviews again, with a condition that only joins with pageview ids that
are after the one displayed for the products page, so we can bring in the session that followed
the products page with the MIN function
GROUP BY
        Prod_page.website_session_id;
-- using the temporary table we produced above, to aggregate for our final click through and
conversion rates
SELECT
    YEAR(prod_created_at) AS yr,
    MONTH(prod_created_at) AS mo,
    COUNT(DISTINCT products_pageview_id) AS product_views,
    COUNT(DISTINCT clicked_through) AS clicked_through,
    ROUND(COUNT(DISTINCT clicked_through) / COUNT(DISTINCT products_pageview_id),
3) AS click_through_rt,
```

```
  ROUND(COUNT(DISTINCT order_id) / COUNT(DISTINCT products_pageview_id), 3) AS
product_to_order_cvr_rate
FROM
        click_throughs ct
LEFT JOIN
        orders o
                USING(website_session_id)
GROUP BY
        1,2;
```

yr	mo	product_views	clicked_through	click_through_rt	product_to_order_cvr_rate
2012	3	743	530	0.713	0.081
2012	4	1447	1029	0.711	0.068
2012	5	1584	1135	0.717	0.068
2012	6	1752	1247	0.712	0.080
2012	7	2018	1438	0.713	0.084
2012	8	3012	2198	0.730	0.076
2012	9	3126	2258	0.722	0.092
2012	10	4030	2948	0.732	0.092
2012	11	6743	4849	0.719	0.092
2012	12	5013	3620	0.722	0.101
2013	1	3380	2595	0.768	0.116
2013	2	3685	2803	0.761	0.135
2013	3	3371	2576	0.764	0.114
2013	4	4362	3356	0.769	0.127
2013	5	4684	3609	0.770	0.122
2013	6	4600	3536	0.769	0.129
2013	7	5020	3890	0.775	0.120
2013	8	5226	3951	0.756	0.116
2013	9	5399	4072	0.754	0.117
2013	10	6038	4564	0.756	0.117
2013	11	7886	5900	0.748	0.109
2013	12	8840	7026	0.795	0.118
2014	1	7790	6387	0.820	0.126
2014	2	7960	6485	0.815	0.128
2014	3	8110	6669	0.822	0.131
2014	4	9744	7958	0.817	0.127
2014	5	10261	8465	0.825	0.133
2014	6	10011	8260	0.825	0.124
2014	7	10837	8958	0.827	0.119
2014	8	10768	8980	0.834	0.123
2014	9	11128	9156	0.823	0.128
2014	10	12335	10235	0.830	0.130
2014	11	14476	12020	0.830	0.137
2014	12	17240	14609	0.847	0.134
2015	1	15217	12992	0.854	0.138
2015	2	14111	11955	0.847	0.144

ANALYSIS OF QUERY 5:

- The most notable thing is that our conversion rate to order has doubled from 6-8% to 14%, through our website improvements, and product additions we have been able to translate this into a higher order and click through rate rate.

- Bringing in a variety of products, and products at a lower price points are some key highlights, as we've gradually and cleverly introduced said products, the click through and order rate has increased

Query 6 :

- Look Into How Our Products Are Cross Selling On Each Other And How That Has Impacted Customer Behavior

*/

```
-- finding the date of when the final product was introduced so we can have a fair assessment of the cross selling stats

SELECT
	MIN(created_at) -- it was '2014-12-05'
FROM
	website_pageviews
WHERE
	pageview_url = '/the-hudson-river-mini-bear' -- the final product introduced
GROUP BY
	pageview_url
;

-- moving onto the main query to produce the final cross sell data below

SELECT
   primary_item,
-- using aggregations from the sub queried data, to give us an overview of the cross selling patterns and behaviors of each product
   COUNT(DISTINCT order_id) AS primary_orders,
   COUNT(CASE WHEN not_primary_1 = 1 THEN 1 ELSE NULL END) AS p1_x_sell,
   ROUND(COUNT(CASE WHEN not_primary_1 = 1 THEN 1 ELSE NULL END)
		/ COUNT(DISTINCT order_id),2) AS p1_x_sell_rt,
   COUNT(CASE WHEN not_primary_2 = 1 THEN 1 ELSE NULL END) AS p2_x_sell,
   ROUND(COUNT(CASE WHEN not_primary_2 = 1 THEN 1 ELSE NULL END)
		/ COUNT(DISTINCT order_id),2) AS p2_x_sell_rt,
   COUNT(CASE WHEN not_primary_3 = 1 THEN 1 ELSE NULL END) AS p3_x_sell,
   ROUND(COUNT(CASE WHEN not_primary_3 = 1 THEN 1 ELSE NULL END)
		/ COUNT(DISTINCT order_id),2) AS p3_x_sell_rt,
   COUNT(CASE WHEN not_primary_4 = 1 THEN 1 ELSE NULL END) AS p4_x_sell,
   ROUND(COUNT(CASE WHEN not_primary_4 = 1 THEN 1 ELSE NULL END)
		/ COUNT(DISTINCT order_id),2) AS p4_x_sell_rt
FROM
	(
```

```
-- creating a sub query so it's possible to aggregate the cross sell rate using 1s & 0s as flags, to decide if the product is the primary purchase or a cross sell
	SELECT
		order_id,
		SUM(CASE WHEN is_primary_item = 1 AND product_id = 1 THEN 1 ELSE 0 END) AS primary_1,
		SUM(CASE WHEN is_primary_item = 1 AND product_id = 2 THEN 1 ELSE 0 END) AS primary_2,
		SUM(CASE WHEN is_primary_item = 1 AND product_id = 3 THEN 1 ELSE 0 END) AS primary_3,
		SUM(CASE WHEN is_primary_item = 1 AND product_id = 4 THEN 1 ELSE 0 END) AS primary_4,
		SUM(CASE WHEN is_primary_item = 0 AND product_id = 1 THEN 1 ELSE 0 END) AS not_primary_1,
		SUM(CASE WHEN is_primary_item = 0 AND product_id = 2 THEN 1 ELSE 0 END) AS not_primary_2,
		SUM(CASE WHEN is_primary_item = 0 AND product_id = 3 THEN 1 ELSE 0 END) AS not_primary_3,
		SUM(CASE WHEN is_primary_item = 0 AND product_id = 4 THEN 1 ELSE 0 END) AS not_primary_4,
		SUM(CASE WHEN is_primary_item = 1 THEN product_id ELSE 0 END) AS primary_item
	FROM
		order_items oi
	WHERE
		created_at > '2014-12-05'
	-- using this timeframe so all products have been selling for the same time period
	GROUP BY
		order_id ) AS flags
GROUP BY
	primary_item
ORDER BY
	primary_item
;
```

primary_item	primary_orders	p1_x_sell	p1_x_sell_rt	p2_x_sell	p2_x_sell_rt	p3_x_sell	p3_x_sell_rt	p4_x_sell	p4_x_sell_rt
1	4467	0	0.00	238	0.05	553	0.12	933	0.21
2	1277	25	0.02	0	0.00	40	0.03	260	0.20
3	929	84	0.09	40	0.04	0	0.00	208	0.22
4	581	16	0.03	9	0.02	22	0.04	0	0.00

ANALYSIS OF QUERY 6:

- Product 1 inspires the most primary orders, potentially due to being a longer lasting staple product with more market awareness however product 4 seems to cause the most

cross sales, likely in being a cheaper product it has higher likelihood of an impulse purchase.

- Through introducing cheaper products we have been able to inspire more sales with our customers
- Going forward if we wanted to experiment or push for more cross sells, it seems potential discounts/promos, or further cheaper products could aid this mission, specifically around the cheaper options

SUMMARY OF ANALYSIS:

- *Overall our business has seen great growth as the analysis shows, from our diligent work at analysing our website and customer behavior patterns and innovation in bringing in new products we have shown great entrepreneurial spirit as a business, and our sales trends speak for themselves*