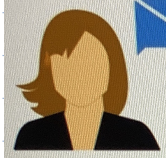

NEW MESSAGE
March, 20 2015

From: **Cindy Sharp** *(CEO)*
Subject: **Need Help w/ Fundraising**

Good morning!

Now that we've been in market for 3 years, we've generated enough growth to raise a much larger round of venture capital funding. We're close to securing a large round from one of the best West Coast firms.

I need your analytical skills to help me paint a picture of high growth, and data-driven performance optimization.

Can you help?
-Cindy

1 **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.**

~ 0:54

```
SELECT
    MIN(DATE(website_session.created_at)) AS quarter_start,
    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
WHERE
    website_sessions.created_at < '2015-01-01'
GROUP BY
    YEAR(website_session.created_at),
    QUARTER(website_session.created_at);
```

created_date	sessions	orders
2012-03-19	1853	59
2012-04-01	11436	347
2012-07-01	16851	682
2012-10-01	32270	1497
2013-01-01	19834	1272
2013-04-01	24727	1712
2013-07-01	27597	1841
2013-10-01	40546	2613
2014-01-01	46714	3071
2014-04-01	53077	3833
2014-07-01	57106	4045
2014-10-01	76399	5906

2 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.

~ 2:40

```
SELECT
    MIN(DATE(website_sessions.created_at)) AS created_date,
    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
    ROUND(COUNT(DISTINCT orders.order_id) /
        COUNT(DISTINCT website_sessions.website_session_id),3)
        AS conv_rate,
    ROUND(SUM(orders.price_usd) / COUNT(DISTINCT orders.order_id),2)
        AS rev_per_order,
    ROUND(SUM(orders.price_usd) /
        COUNT(DISTINCT website_sessions.website_session_id),2)
        AS rev_per_session
FROM website_sessions
  LEFT JOIN orders
      ON website_sessions.website_session_id = orders.website_session_id
WHERE
    website_sessions.created_at < '2015-01-01'
GROUP BY
    YEAR(website_session.created_at),
    QUARTER(website_session.created_at);
```

created_date	sessions	conversion_rate	revenue_per_order	revenue_per_session
2012-03-19	1853	0.032	49.99	1.59
2012-04-01	11436	0.030	49.99	1.52
2012-07-01	16851	0.040	49.99	2.02
2012-10-01	32270	0.046	49.99	2.32
2013-01-01	19834	0.064	52.14	3.34
2013-04-01	24727	0.069	51.54	3.57
2013-07-01	27597	0.067	51.73	3.45
2013-10-01	40546	0.064	54.69	3.52
2014-01-01	46714	0.066	62.16	4.09
2014-04-01	53077	0.072	64.36	4.65
2014-07-01	57106	0.071	64.47	4.57
2014-10-01	76399	0.077	63.83	4.93

3 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?

~ 5:27

```
SELECT
    MIN(DATE( website_sessions.created_at) AS created_date,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND
        utm_source = 'gsearch' THEN orders.order_id ELSE NULL END) AS gsearch_nonbrand,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND
        utm_source = 'bsearch' THEN orders.order_id ELSE NULL END) AS bsearch_nonbrand,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN
        orders.order_id ELSE NULL END) AS paid_brand,
    COUNT(DISTINCT CASE WHEN utm_source IS NULL AND http_referer IS NULL
        THEN orders.order_id ELSE NULL END) AS direct_type_in,
    COUNT(DISTINCT CASE utm_source IS NULL AND http_referer IS NOT NULL
        THEN orders.order_id ELSE NULL END) AS organic_search,
FROM website_sessions
    LEFT JOIN orders
        ON website_sessions.website_session_id = orders.website_session_id
WHERE
    website_sessions.created_at < '2015-01-01'
GROUP BY
    YEAR (website_session.created_at),
    QUARTER(website_session.created_at);
```

created_date	gsearch_nonbrand	bsearch_nonbrand	brand_overall	organic_search	direct_type_in
2012-03-19	59	0	0	0	0
2012-04-01	292	0	19	15	21
2012-07-01	479	82	49	40	32
2012-10-01	916	311	87	94	89
2013-01-01	764	183	109	125	91
2013-04-01	1111	236	112	134	119
2013-07-01	1133	246	155	166	141
2013-10-01	1656	291	247	221	198
2014-01-01	1667	344	353	340	312
2014-04-01	2200	423	410	434	366
2014-07-01	2267	435	431	448	402
2014-10-01	3244	686	616	602	532

4 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.

~ 8:45

```
SELECT
    MIN(DATE(website_sessions.created_at) AS created_date,
    COUNT(DISTINCT website_session.website_session_id) AS sessions
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND
            utm_source = 'gsearch' THEN orders.order_id ELSE NULL END) /
            COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND
            utm_source = 'gsearch' THEN website_session.website_session_id ELSE NULL END)
            AS gsearch_nonbrand_conv_rt,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND
            utm_source = 'bsearch' THEN orders.order_id ELSE NULL END) /
            COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' AND
            utm_source = 'bsearch' THEN website_session.website_session_id ELSE NULL END)
            AS bsearch_nonbrand_conv_rate,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN
            orders.order_id ELSE NULL END) /
            COUNT(DISTINCT CASE WHEN utm_campaign = 'brand'
            THEN website_session.website_session_id ELSE NULL END)
            AS brand_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_session.website_session_id  ELSE NULL END)
            AS direct_type_in_conv_rt,
    COUNT(DISTINCT CASE utm_source IS NULL AND http_referer IS NOT NULL
            THEN orders.order_id ELSE NULL END) /
            COUNT(DISTINCT CASE utm_source IS NULL AND http_referer IS NOT NULL
                    THEN website_session.website_session_id  ELSE NULL END)
            AS organic_search_conv_rt
FROM website_sessions
    LEFT JOIN orders
        ON website_sessions.website_session_id = orders.website_session_id
WHERE
    website_sessions.created_at < '2015-01-01'
GROUP BY
    YEAR (website_session.created_at),
    QUARTER(website_session.created_at);
```

created_date	sessions	gsearch_nonbrand_conv_rt	bsearch_nonbrand_conv_rt	brand_overall_conv_rt	organic_search_conv_rt	direct_type_in_conv_rt
2012-03-19	1853	0.0323	NULL	0.0000	0.0000	0.0000
2012-04-01	11436	0.0285	NULL	0.0503	0.0359	0.0538
2012-07-01	16851	0.0382	0.0411	0.0615	0.0499	0.0444
2012-10-01	32270	0.0437	0.0496	0.0528	0.0541	0.0538
2013-01-01	19834	0.0611	0.0696	0.0706	0.0751	0.0611
2013-04-01	24727	0.0684	0.0687	0.0668	0.0760	0.0737
2013-07-01	27597	0.0642	0.0701	0.0713	0.0734	0.0709
2013-10-01	40546	0.0629	0.0600	0.0798	0.0686	0.0652
2014-01-01	46714	0.0694	0.0707	0.0839	0.0762	0.0768
2014-04-01	53077	0.0700	0.0689	0.0804	0.0795	0.0737
2014-07-01	57106	0.0705	0.0699	0.0755	0.0738	0.0704
2014-10-01	76399	0.0781	0.0844	0.0814	0.0780	0.0747

5 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.

11:18

למשוך לפי חודש מגמת הכנסה לפי מוצר ומרג'ין לפי מוצר, סה כולל
של מכירות וסה כולל הכנסה.

```
SELECT
    YEAR(created_at),
    MONTH(created_at),
    SUM(CASE WHEN product_id=1 THEN price_usd ELSE NULL END) AS mrfuzzy_revenue,
    ROUND(SUM(WHEN product_id=1 THEN price_usd - cogs_usd ELSE NULL END),1) AS mrfuzzy_margin,
    SUM(CASE WHEN product_id=2 THEN price_usd ELSE NULL END) AS lovebear_revenue,
    ROUND(SUM(WHEN product_id=2 THEN price_usd - cogs_usd ELSE NULL END),1) AS lovebear_margin,
    SUM(CASE WHEN product_id=3 THEN price_usd ELSE NULL END) AS birthday_revenue,
    ROUND(SUM(WHEN product_id=3 THEN price_usd - cogs_usd ELSE NULL END),1) AS birthday_margin,
    SUM(CASE WHEN product_id=4 THEN price_usd ELSE NULL END) AS mini_revenue
    ROUND(SUM(WHEN product_id=4 THEN price_usd - cogs_usd ELSE NULL END),1) AS mini_margin,
    SUM(price_usd) AS total_revenue,
    COUNT(DISTINCT order_id) AS total_sales
FROM order_item
WHERE
    created_at < '2015-01-01'
GROUP BY
    YEAR(created_at),
    MONTH(created_at)
```

	created_month	mrfuzzy_revenue	mrfuzzy_margin	love_bear_revenue	love_bear_margin	birthday_bear_revenue	birthday_bear_margin	mini_bear_revenue	mini_bear_margin	total_revenue	total_sales
012	3	2949.41	1799.5	NULL	NULL	NULL	NULL	NULL	NULL	2949.41	59
012	4	4999.00	3050.0	NULL	NULL	NULL	NULL	NULL	NULL	4999.00	100
012	5	5298.94	3233.0	NULL	NULL	NULL	NULL	NULL	NULL	5298.94	106
012	6	7048.59	4300.5	NULL	NULL	NULL	NULL	NULL	NULL	7048.59	141
012	7	8398.32	5124.0	NULL	NULL	NULL	NULL	NULL	NULL	8398.32	168
012	8	11447.71	6984.5	NULL	NULL	NULL	NULL	NULL	NULL	11447.71	229
012	9	14247.15	8692.5	NULL	NULL	NULL	NULL	NULL	NULL	14247.15	285
012	10	18296.34	11163.0	NULL	NULL	NULL	NULL	NULL	NULL	18296.34	366
2012	11	31093.78	18971.0	NULL	NULL	NULL	NULL	NULL	NULL	31093.78	622
2012	12	25444.91	15524.5	NULL	NULL	NULL	NULL	NULL	NULL	25444.91	509
2013	1	17046.59	10400.5	2759.54	1725.0	NULL	NULL	NULL	NULL	19806.13	387
2013	2	16496.70	10065.0	9718.38	6075.0	NULL	NULL	NULL	NULL	26215.08	492
2013	3	16396.72	10004.0	3899.35	2437.5	NULL	NULL	NULL	NULL	20296.07	393
2013	4	22645.47	13816.5	5579.07	3487.5	NULL	NULL	NULL	NULL	28224.54	546
2013	5	24695.06	15067.0	4919.18	3075.0	NULL	NULL	NULL	NULL	29614.24	576
2013	6	24995.00	15250.0	5399.10	3375.0	NULL	NULL	NULL	NULL	30394.10	590
2013	7	25294.94	15433.0	5819.03	3637.5	NULL	NULL	NULL	NULL	31113.97	603
2013	8	25694.86	15677.0	5879.02	3675.0	NULL	NULL	NULL	NULL	31573.88	612
2013	9	26794.64	16348.0	5759.04	3600.0	NULL	NULL	NULL	NULL	32553.68	626
2013	10	29794.04	18178.0	8038.66	5025.0	NULL	NULL	NULL	NULL	37832.70	700
2013	11	36742.65	22417.5	10618.23	6637.5	NULL	NULL	NULL	NULL	47360.88	875
2013	12	40541.89	24735.5	10738.21	6712.5	6392.61	4378.5	NULL	NULL	57672.71	1037
2014	1	36592.68	22326.0	11218.13	7012.5	9106.02	6237.0	NULL	NULL		

sult 8 ×

tput

6 14:53

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.

להבין קצת לפני תוצאים.

עמודה של כניסות לעמוד בדיקה

עמודה של הקלקות לעמוד הבא (בתהליכים - ועני חלקי כניסות למוצרים)

עמודה 3 פתחה להזמנה אחוזים.

```
-- STEP 1: Pull all relevant sessions
CREATE TEMPORARY TABLE products_sessions
SELECT
        YEAR(created_at) AS yr,
        MONTH(created_at) AS mo,
        website_session_id,
        website_pageview_id,
FROM website_pageviews
WHERE
       created_at < '2013-01-01',
       pageview_url = '/products' );


SELECT
        products_session.yr,
        products_session.mo,
        COUNT(DISTINCT products_session.website_session_id) AS sessions,
        ROUND(COUNT(DISTINCT website_pageviews.website_session_id) /
             COUNT(DISTINCT products_session.website_session_id), 2) AS clicked_through_rt,
        ROUND(COUNT(DISTINCT orders.order_id) /
             COUNT(DISTINCT products_session.website_session_id), 2) AS conv_rt_products_to_order
FROM products_session
   LEFT JOIN website_pageviews
       ON products_session.website_session_id = website_pageviews.website_session_id
       AND products_session.website_pageview_id < website_pageviews.website_pageview_id
       LEFT JOIN orders
             ON orders.website_session_id = products_session.website_session_id
GROUP BY
        products_session.yr,
        products_session.mo;
```

yr	mo	sessions	clicked_through_tr	conv_rt_product_to_order
2012	3	729	0.71	0.08
2012	4	1441	0.71	0.07
2012	5	1575	0.72	0.07
2012	6	1770	0.71	0.08
2012	7	2001	0.71	0.08
2012	8	3006	0.73	0.08
2012	9	3132	0.72	0.09
2012	10	4011	0.73	0.09
2012	11	6756	0.72	0.09
2012	12	5011	0.72	0.10
2013	1	3370	0.77	0.11
2013	2	3682	0.76	0.13
2013	3	3386	0.77	0.12
2013	4	4329	0.77	0.13
2013	5	4700	0.77	0.12
2013	6	4602	0.77	0.13
2013	7	4997	0.78	0.12
2013	8	5271	0.76	0.12
2013	9	5349	0.76	0.12
2013	10	6051	0.75	0.12
2013	11	7906	0.75	0.11
2013	12	8802	0.79	0.12
2014	1	7793	0.82	0.13
2014	2	7962	0.82	0.13
2014	3	8073	0.82	0.13
2014	4	9708	0.82	0.13
2014	5	10340	0.82	0.13

Result 16