



NEW MESSAGE

November 27, 2012

From: Cindy Sharp (CEO)

Subject: Board Meeting Next Week

Good morning,

I need some help preparing a presentation for the board meeting next week.

The board would like to have a better understanding of our growth story over our first 8 months. This will also be a good excuse to show off our analytical capabilities a bit.

-Cindy

1

Gsearch seems to be the biggest driver of our business. Could you pull monthly trends for gsearch sessions and orders so that we can showcase the growth there?

SELECT

```
MIN(DATE(web site_sessions.created_at)) AS start_month,  
COUNT(DISTINCT CASE WHEN web site_sessions.Utm_Source = 'gsearch' THEN  
web site_sessions.website_session_id ELSE NULL END) AS sessions,  
COUNT(DISTINCT CASE WHEN Orders.order_id THEN  
Orders.website_session_id) AS orders
```

FROM WebSiteSessions

LEFT JOIN Orders

ON WebSiteSessions.website_session_id = Orders.website_session_id

WHERE

WebSiteSessions.created_at < '2012-11-27'

AND web site_sessions.Utm_Source = 'gsearch'

GROUP BY

YEAR(web site_sessions.created_at),

MONTH(web site_sessions.created_at)

start_month	sessions	orders
2012-03-19	1834	59
2012-04-01	3559	93
2012-05-01	3398	95
2012-06-01	3610	122
2012-07-01	3773	144
2012-08-01	4896	185
2012-09-01	4490	186
2012-10-01	5504	236
2012-11-01	8421	351

2

Next, it would be great to see a similar monthly trend for Gsearch, but this time **splitting out nonbrand and brand campaigns separately**. I am wondering if brand is picking up at all. If so, this is a good story to tell.

SELECT

```

MIN(DATE(wbs_sessions.created_at)) AS month_start,
COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN
wbs_sessions.website_session_id ELSE NULL END) AS nonbrand_sessions,
COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN
wbs_sessions.website_session_id ELSE NULL END) AS brand_sessions
COUNT(DISTINCT CASE WHEN orders.orderid AND utm_campaign = 'nonbrand' THEN
orders.website_session_id ELSE NULL END) AS nonbrand_orders,
COUNT(DISTINCT CASE WHEN orders.orderid AND utm_campaign = 'brand' THEN
orders.website_session_id ELSE NULL END) AS brand_orders,
FROM wbs_sessions
LEFT JOIN orders
ON wbs_sessions.website_session_id = orders.website_session_id
WHERE
wbs_sessions.created_at > '2012-11-27'
AND utm_source = 'gsearch'
AND utm_campaign IN ('nonbrand', 'brand')
GROUP BY
YEAR(website_sessions.created_at),
MONTH(website_sessions.created_at)

```

month_start	nonbrand_sessions	brand_sessions	nonbrand_orders	brand_orders
2012-03-19	1826	8	59	0
2012-04-01	3497	62	87	6
2012-05-01	3283	115	89	6
2012-06-01	3470	140	116	6
2012-07-01	3622	151	134	10
2012-08-01	4693	203	175	10
2012-09-01	4225	265	170	16
2012-10-01	5172	332	221	15
2012-11-01	8048	373	334	17

3

While we're on Gsearch, could you dive into nonbrand, and pull **monthly sessions and orders split by device type**? I want to flex our analytical muscles a little and show the board we really know our traffic sources.

SELECT

```

MIN(DATE(wbs.tb_sessions.created_at)) AS month_start,
COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN
wbs.tb_sessions.website_session_id ELSE NULL END) AS desktop_sessions,
COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN
wbs.tb_sessions.website_session_id ELSE NULL END) AS mobile_sessions,
COUNT(DISTINCT CASE WHEN device_type = 'nonbrand' THEN
orders.order_id ELSE NULL END) AS nonbrand_orders,
COUNT(DISTINCT CASE WHEN device_type = 'brand' THEN
orders.order_id ELSE NULL END) AS brand_orders,
FROM wbs.tb_sessions
LEFT JOIN orders
ON wbs.tb_sessions.website_session_id = orders.website_session_id

```

WHERE

wbs.tb_sessions.created_at < '2012-11-27'

AND utm_source = 'gsearch'

AND utm_campaign = 'nonbrand'

GROUP BY

YEAR(wbs.tb_sessions.created_at),

MONTH(wbs.tb_sessions.created_at)

month_start	desktop_sessions	mobile_sessions	desktop_orders	mobile_orders
2012-03-19	1116	710	49	10
2012-04-01	2132	1365	76	11
2012-05-01	2253	1030	81	8
2012-06-01	2698	772	108	8
2012-07-01	2751	871	120	14
2012-08-01	3526	1167	166	9
2012-09-01	3170	1055	154	16
2012-10-01	3919	1253	202	19
2012-11-01	6112	1936	303	31

4

I'm worried that one of our more pessimistic board members may be concerned about the large % of traffic from Gsearch. Can you pull **monthly trends for Gsearch, alongside monthly trends for each of our other channels?**

SELECT

MIN(DATE(wbsitB_sessions.CheatB_at)) AS month_start,
COUNT(DISTINCT CASE WHEN web_site_sessions.Utm_Source = 'gsearch' THEN
web_site_sessions.website_session_id ELSE NULL END) AS sessions,
COUNT(DISTINCT CASE WHEN web_site_sessions.Utm_Source = 'b search' THEN
web_site_sessions.website_session_id ELSE NULL END) AS sessions,
COUNT(DISTINCT CASE WHEN web_site_sessions.Utm_Source IS NULL AND http_Referrer
IS NOT NULL THEN web_site_sessions.website_session_id ELSE NULL END) AS organic_search_sessions
COUNT(DISTINCT CASE WHEN web_site_sessions.Utm_Source IS NULL AND http_Referrer
IS NULL THEN web_site_sessions.website_session_id ELSE NULL END) AS direct_type_in_sessions
FROM wbsitB_sessions
LEFT JOIN otcas
ON wbsitB_sessions.website_session_id = otcas.website_session_id

WHERE

Wk6s_itb_sessions. (1)part02 - at 2 '2012-11-27'

GROUP BY

YEAR (`web site_sessions.created-at`),
MONTH (`web site_sessions.created-at`)

month_start	gsearch_sessions	bsearch_sessions	organic_search_sessions	direct_type_in_sessions
2012-03-19	1834	2	8	
2012-04-01	3559	11	75	9
2012-05-01	3398	25	147	70
2012-06-01	3610	25	196	149
2012-07-01	3773	44	203	171
2012-08-01	4896	687	265	188
2012-09-01	4490	1440	333	248
2012-10-01	5504	1768	427	284
2012-11-01	8421	2705	520	443
				468

5

I'd like to tell the story of our website performance improvements over the course of the first 8 months.
 Could you pull session to order conversion rates, by month?

SELECT

```
MIN(DATE website_sessions.created_at) AS month_start
COUNT(DISTINCT website_sessions.website_session_id) AS sessions
COUNT(DISTINCT orders.order_id) AS orders
COUNT(DISTINCT orders.order_id) / COUNT(DISTINCT website_sessions.website_session_id)
```

AS session_to_order_rt

FROM website_sessions

LEFT JOIN orders

ON website_sessions.website_session_id = orders.website_session_id

WHERE

website_sessions.created_at < '2012-11-27'

GROUP BY

YEAR(website_sessions.created_at),
 MONTH(website_sessions.created_at)

month_start	sessions	orders	session_to_order_rt
2012-03-19	1853	59	3.1840
2012-04-01	3715	100	2.6918
2012-05-01	3719	106	2.8502
2012-06-01	4002	141	3.5232
2012-07-01	4208	168	3.9924
2012-08-01	6096	229	3.7566
2012-09-01	6547	285	4.3531
2012-10-01	8142	366	4.4952
2012-11-01	12114	535	4.4164

For the gsearch lander test, please estimate the revenue that test earned us (*Hint: Look at the increase in CVR from the test (Jun 19 - Jul 28), and use nonbrand sessions and revenue since then to calculate incremental value*)

-- STEP1: find out when the page '/lander-1' launched

SELECT

MIN(click_order) AS first_click_order,

MIN(website_pageview_id) AS first_pageview

FROM website_sessions

WHERE

pageview_url = '/lander-1'

-- first click_order = 2012-06-19

-- first pageview = 23504

-- STEP2: finding the first website_pageview_id for the relevant sessions

CREATE TEMPORARY TABLE first_pageview_w_landing_page

SELECT

website_sessions.website_session_id

MIN(website_pageviews.website_pageview_id) AS first_pageview

website_pageviews.pageview_url AS landing_page

From website_sessions

LEFT JOIN website_pageviews

ON website_sessions.website_session_id = website_pageviews.website_session_id

WHERE

website_sessions.click_order BETWEEN '2012-06-19' AND '2012-07-28'

AND website_pageviews.website_pageview_id > '23504'

AND website_pageviews.pageview_url IN ('/home', '/lander-1')

AND utm_source = 'gsearch'

AND utm_campaign = 'nonbrand'

GROUP BY

website_sessions.website_session_id

-- STEP3: combine with orders table and find the orders % from each landing_page

SELECT

first_pageview_w_landing_page.landing_page

COUNT(DISTINCT first_pageview_w_landing_page.website_session_id) AS sessions

COUNT(DISTINCT orders.order_id) AS orders

COUNT(DISTINCT orders.order_id) / COUNT(DISTINCT first_pageview_w_landing_page.website_session_id) * 100

AS conversion_pct

From first_pageview_w_landing_page

LEFT JOIN orders

ON first_pageview_w_landing_page.website_session_id = orders.website_session_id

GROUP BY

first_pageview_w_landing_page.landing_page

landing_page	sessions	orders	conversion_rate
/home	2241	71	3.1682
/lander-1	2296	94	4.0941

-- we have 0.00926 additional orders per session on landing page (0.00941 - 0.031682)

-- STEP 4: finding the most recent pageview for gsearch where the traffic sent to home and how many sessions we have since that last

SELECT

MAX(website_sessions.website_session_id)
AS most_recent_gsearch_nonbrand_home_pageview

FROM website_sessions

LEFT JOIN website_pageviews

ON website_sessions.website_session_id = website_pageviews.website_session_id

WHERE

utm_source = 'gsearch'
AND utm_campaign = 'nonbrand'
AND pageview_url = 'home'
AND website_sessions.created_at < '2012-11-27';

-- max website_session_id = 17145

SELECT

COUNT(website_session_id) AS sessions_since_last

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')

-- the number of sessions is 22450

-- and how we just need to calculate the incremental value

-- we are going to multiply the number of sessions that we found (22450) by the additional orders

-- per session (0.0926) - 22450 * 0.0926 ≈ 207

-- so roughly in month, we have approximately 50 extra order per month!

STEP1: create flags table

CREATE TEMPORARY TABLE flags

SELECT

```
WEBSITE_SESSIONS, WEBSITE_SESSION_ID,
WEBSITE_PAGEVIEWS, PAGEVIEW_URL,
WEBSITE_PAGEVIEWS.CREATED_AT AS PAGEVIEW_CREATED_AT,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/home' THEN 1 ELSE 0 END AS home_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/lander1' THEN 1 ELSE 0 END AS lander1_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/products' THEN 1 ELSE 0 END AS product_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END
AS mfuzzy_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/cart' THEN 1 ELSE 0 END AS cart_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/shipping' THEN 1 ELSE 0 END AS shipping_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/billing' THEN 1 ELSE 0 END AS billing_page,
CASE WHEN WEBSITE_PAGEVIEWS.PAGEVIEW_URL = '/thank-you-for-your-order' THEN 1 ELSE 0 END
AS thank_you_page
```

FROM WEBSITE_SESSIONS

LEFT JOIN WEBSITE_PAGEVIEWS

ON WEBSITE_SESSIONS.WEBSITE_SESSION_ID = WEBSITE_PAGEVIEWS.WEBSITE_SESSION_ID

WHERE

WEBSITE_SESSIONS.CREATED_AT BETWEEN '2012-06-19' AND '2012-07-28'

AND utm_source = 'search'

AND utm_campaign = 'nonbrand'

ORDER BY

WEBSITE_SESSIONS.WEBSITE_SESSION_ID,

WEBSITE_PAGEVIEWS.CREATED_AT;

website_session_id	pageview_url	pageview_created_at	home_page	lander1_page	products_page	mfuzzy_page	cart_page	shipping_page	billing_page	thank_you_page
11638	/home	2012-06-19 00:03:32	1	0	0	0	0	0	0	0
11639	/home	2012-06-19 00:13:27	1	0	0	0	0	0	0	0
11639	/products	2012-06-19 00:15:34	0	0	1	0	0	0	0	0
11639	/the-original-mr-fuzzy	2012-06-19 00:19:19	0	0	0	0	1	0	0	0
11639	/cart	2012-06-19 00:23:12	0	0	0	0	0	1	0	0
11640	/home	2012-06-19 00:21:02	1	0	0	0	0	0	0	0
11640	/products	2012-06-19 00:22:42	0	0	1	0	0	0	0	0
11640	/the-original-mr-fuzzy	2012-06-19 00:26:56	0	0	0	1	0	0	0	0
11641	/home	2012-06-19 00:21:06	1	0	0	0	0	0	0	0
11641	/products	2012-06-19 00:22:27	0	0	1	0	0	0	0	0
11641	/the-original-mr-fuzzy	2012-06-19 00:23:16	0	0	0	1	0	0	0	0
11642	/home	2012-06-19 00:22:07	1	0	0	0	0	0	0	0
11642	/products	2012-06-19 00:22:36	0	0	1	0	0	0	0	0
11643	/home	2012-06-19 00:29:51	1	0	0	0	0	0	0	0
11643	/products	2012-06-19 00:32:53	0	0	1	0	0	0	0	0
11643	/the-original-mr-fuzzy	2012-06-19 00:37:18	0	0	0	1	0	0	0	0
11643	/cart	2012-06-19 00:40:30	0	0	0	0	1	0	0	0
11643	/shipping	2012-06-19 00:41:56	0	0	0	0	0	1	0	0
11643	/billing	2012-06-19 00:43:31	0	0	0	0	0	0	1	0
11645	/home	2012-06-19 00:31:32	1	0	0	0	0	0	0	0
11645	/products	2012-06-19 00:35:25	0	0	1	0	0	0	0	0
11646	/home	2012-06-19 00:39:38	1	0	0	0	0	0	0	0
11647	/home	2012-06-19 00:50:12	1	0	0	0	0	0	0	0

-- STEP 2: Create the site route for each session_id

CREATE TEMPORARY TABLE websitb_pathway

SELECT

websitb_session_id,
MAX(home_page) AS home_landing_page,
MAX(lander1_page) AS landing_landing_page,
MAX(products_page) AS product_made_it,
MAX(mtfuzzx_page) AS mtfuzzx_made_it,
MAX(cart_page) AS cart_made_it,
MAX(shipping_page) AS shipping_made_it,
MAX(billing_page) AS billing_made_it,
MAX(thank_you_page) AS thank_you_made_it

From flags

GROUP BY

websitb_session_id;

website_session_id	home_landing_page	lander1_landing_page	product_made_it	mrfuzzy_made_it	cart_made_it	shipping_made_it	billing_made_it	thank_you_made_it
11638	1	0	0	0	0	0	0	0
11639	1	0	1	1	1	0	0	0
11640	1	0	1	1	0	0	0	0
11641	1	0	1	1	0	0	0	0
11642	1	0	1	1	0	0	0	0
11643	1	0	1	0	0	0	0	0
11645	1	0	1	1	1	1	0	0
11646	1	0	1	0	0	0	1	0
11647	1	0	0	0	0	0	0	0
11648	1	0	0	0	0	0	0	0
11649	1	0	1	1	0	0	0	0
11650	1	0	1	0	0	0	0	0
11653	1	0	1	1	0	0	0	0
11654	1	0	0	0	0	0	0	0
11655	1	0	1	0	0	0	0	0
11656	1	0	1	1	1	0	0	0
11659	1	0	0	0	0	0	0	0
11660	1	0	1	1	0	0	0	0
11661	1	0	1	1	0	0	0	0
11662	1	0	1	0	0	0	0	0
11663	1	0	1	1	0	0	0	0
11664	1	0	1	1	0	0	0	0
11665	1	0	1	1	0	0	0	0

SELECT

CASE

WHEN home_landing_page = 1 THEN 'saw_home_page'

WHEN lander1_landing_page = 1 THEN 'saw_lander_page'

ELSE 'check logic'

END AS segment

COUNT(DISTINCT websitb_session_id) AS sessions

COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN

websitb_session_id ELSE NULL END) AS to_product

COUNT(DISTINCT CASE WHEN mtfuzzx_made_it = 1 THEN

websitb_session_id ELSE NULL END) AS to_mtfuzzx

COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN

websitb_session_id ELSE NULL END) AS to_cart

COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN

websitb_session_id ELSE NULL END) AS to_shipping

COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN

websitb_session_id ELSE NULL END) AS to_billing

COUNT(DISTINCT CASE WHEN thank_you_made_it = 1 THEN

websitb_session_id ELSE NULL END) AS to_thank_you

FROM websitb_pathway

GROUP BY

segment;

segment	sessions	product_page	mrfuzzy_page	cart_page	shipping_page	billing_page	thank_you_page
saw_home_page	2277	958	694	299	201	170	73
saw_lander_page	2296	1076	769	346	230	196	94

SELECT

CASE

WHEN home_landing_page, =1 THEN 'saw_home_page'
WHEN lander_landing_page, =1 THEN 'saw_lander_page'
ELSE 'check logic'

END AS segment

COUNT(DISTINCT web_sessions.session_id) AS sessions

COUNT(DISTINCT CASE WHEN product_math_it = 1 THEN
website_session_id ELSE NULL END) / COUNT(DISTINCT web_sessions.session_id)
AS products_rate,

COUNT(DISTINCT CASE WHEN mr_fuzzy_math_it = 1 THEN
website_session_id ELSE NULL END) /
COUNT(DISTINCT CASE WHEN product_math_it = 1 THEN

website_session_id ELSE NULL END) AS mr_fuzzy_rate,

COUNT(DISTINCT CASE WHEN cart_math_it = 1 THEN
website_session_id ELSE NULL END)

COUNT(DISTINCT CASE WHEN mr_fuzzy_math_it = 1 THEN
website_session_id ELSE NULL END) AS cart_rate,

COUNT(DISTINCT CASE WHEN shipping_math_it = 1 THEN
website_session_id ELSE NULL END) /

COUNT(DISTINCT CASE WHEN cart_math_it = 1 THEN
website_session_id ELSE NULL END) AS shipping_rate,

COUNT(DISTINCT CASE WHEN billing_math_it = 1 THEN
website_session_id ELSE NULL END) /

COUNT(DISTINCT CASE WHEN shipping_math_it = 1 THEN
website_session_id ELSE NULL END) AS billing_rate,

COUNT(DISTINCT CASE WHEN thank_you_math_it = 1 THEN
website_session_id ELSE NULL END) /

COUNT(DISTINCT CASE WHEN billing_math_it = 1 THEN
website_session_id ELSE NULL END) AS thank_you_rate

FROM website_pathway

GROUP BY

segment;

segment	products_rate	mr_fuzzy_rate	cart_rate	shipping_rate	billing_rate	thank_you_rate
saw_home_page	0.4207	0.7244	0.4308	0.6722	0.8458	0.4294
saw_lander_page	0.4686	0.7147	0.4499	0.6647	0.8522	0.4796

8

I'd love for you to quantify the impact of our billing test, as well. Please analyze the lift generated from the test (Sep 10 – Nov 10), in terms of revenue per billing page session, and then pull the number of billing page sessions for the past month to understand monthly impact.

לפנינו מושג אחד אחד, כמוBilling. רצוננו להבין
האם יש לנו יותר או פחות סטודנטים מאשר שוטר
במשך תקופה מסוימת, כמו שוטר. וכאן שוטר הוא שוטר שוטר.

```
CREATE TEMPORARY TABLE billing - Page - Sessions - v_07days
```

SELECT

WEGSICHER - PageViews. WEGSICHER - SESSION - id,
WEGSICHER - PageViews. PageVIEW-URL,
OTJOTS. OTJOT - id,
OTJOTS. PATICL - USD

From ~~WEGSIEB~~ - Page 91

LEFT JOIN OTHERS

ON `WBSItem - Project.WBSItem-Session_id = Objets.WBSItem-Session_id`

WHERE

WEGSIEB - Paragraphs created BETWEEN '2012-09-10' AND '2012-11-10'

WBG5iEB - Paßfrikt. WCBG5iEB - Paßfrikt. WCBG5iEB - Paßfrikt. WCBG5iEB - Paßfrikt. WCBG5iEB - Paßfrikt.

WEGSIEB - PAFBRICKS. PAFBRICKS-URL IN ('Billing', 'Billing-2')

SELECT

PasGvich - CHI

COUNT(DISTINCT user_id - session_id) AS sessions

COUNT(DISTINCT Objet_id) AS Objets

`COUNT(DISTINCT order_id) / COUNT(DISTINCT website_session_id)`

AS billing - part - session-to - odd - half

`SUM(P2id_cst) / COUNT(DISTINCT webvisib - session_id) AS krennub_P2t-billing_Page_seeh`

FROM Billing - Page - Sessions - v - updates

GROUP BY

Pag 8 Vichy - CHI;

Result Grid					
	pageview_url	sessions	orders	billing_page_session_to_order	revenue_per_billing_page_seen
>	/billing	655	300	0.4580	22.896183
	/billing-2	652	409	0.6273	31.358758

-- An increase of almost 17%!

-- log more objects!

— An increase of $\approx 9\%$ for the billing people

SELECT

COUNT(DISTINCT WebSiteSessionID) AS Sessions

From website - Page rights

WHERE

CHARTS-
BETWEEN '2012-10-27' AND '2012-11-27'

AND pagelink-uri IN ('/billings', '/billings-2')

-- In the last month we had 1137 sessions

$$-- 1137 \cdot 8.31 = 9539$$

- the monthly impact for the past month is 9539\$!

לעתות קיימת אפשרות לשלוח את הלקוח לאישור בBILLING (בילינג) או לשלוחו ישירות ללקוח. במקרה השני, הלקוח יראה את הלקוח כלקוח פרטי.