

```

2  /*
3  1. Gsearch seems to be the biggest driver of our business. Could you pull monthly
4  trends for gsearch sessions and orders so that we can showcase the growth there?
5  */
6
7  • SELECT
8      YEAR(website_sessions.created_at) AS yr,
9      MONTH(website_sessions.created_at) AS mo,
10     COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
11     COUNT(DISTINCT orders.order_id) AS orders,
12     COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conv_rate
13 FROM website_sessions
14     LEFT JOIN orders
15         ON orders.website_session_id = website_sessions.website_session_id
16 WHERE website_sessions.created_at < '2012-11-27'
17        AND website_sessions.utm_source = 'gsearch'
18 GROUP BY 1,2;
19
20 /*
21 2. Next, it would be great to see a similar monthly trend for Gsearch, but this time splitting out nonbrand
22 and brand campaigns separately. I am wondering if brand is picking up at all. If so, this is a good story to tell.
23 */
24
25 • SELECT
26     YEAR(website_sessions.created_at) AS yr,
27     MONTH(website_sessions.created_at) AS mo,
28     COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN website_sessions.website_session_id ELSE NULL END) AS nonbrand_sessions,
29     COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN orders.order_id ELSE NULL END) AS nonbrand_orders,
30     COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN website_sessions.website_session_id ELSE NULL END) AS brand_sessions,
31     COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN orders.order_id ELSE NULL END) AS brand_orders
32 FROM website_sessions
33     LEFT JOIN orders
34         ON orders.website_session_id = website_sessions.website_session_id
35 WHERE website_sessions.created_at < '2012-11-27'
36        AND website_sessions.utm_source = 'gsearch'
37 GROUP BY 1,2;
38
39
40 /*
41 3. While we're on Gsearch, could you dive into nonbrand, and pull monthly sessions and orders split by device type?
42 I want to flex our analytical muscles a little and show the board we really know our traffic sources.

```

```

45 • SELECT
46     YEAR(website_sessions.created_at) AS yr,
47     MONTH(website_sessions.created_at) AS mo,
48     COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN website_sessions.website_session_id ELSE NULL END) AS desktop_sessions,
49     COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN orders.order_id ELSE NULL END) AS desktop_orders,
50     COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN website_sessions.website_session_id ELSE NULL END) AS mobile_sessions,
51     COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN orders.order_id ELSE NULL END) AS mobile_orders
52 FROM website_sessions
53     LEFT JOIN orders
54         ON orders.website_session_id = website_sessions.website_session_id
55 WHERE website_sessions.created_at < '2012-11-27'
56     AND website_sessions.utm_source = 'gsearch'
57     AND website_sessions.utm_campaign = 'nonbrand'
58 GROUP BY 1,2;
59 /*
60 4. I'm worried that one of our more pessimistic board members may be concerned about the large % of traffic from Gsearch.
61 Can you pull monthly trends for Gsearch, alongside monthly trends for each of our other channels?
62 */
63
64 -- first, finding the various utm sources and referers to see the traffic we're getting
65
66 • SELECT DISTINCT
67     utm_source,
68     utm_campaign,
69     http_referer
70 FROM website_sessions
71 WHERE website_sessions.created_at < '2012-11-27';
72
73
74 • SELECT
75     YEAR(website_sessions.created_at) AS yr,
76     MONTH(website_sessions.created_at) AS mo,
77     COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' THEN website_sessions.website_session_id ELSE NULL END) AS gsearch_paid_sessions,
78     COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' THEN website_sessions.website_session_id ELSE NULL END) AS bsearch_paid_sessions,
79     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_sessions,
80     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_sessions
81 FROM website_sessions
82     LEFT JOIN orders
83         ON orders.website_session_id = website_sessions.website_session_id
84 WHERE website_sessions.created_at < '2012-11-27'
85 GROUP BY 1,2;
86

```

```

94 • SELECT
95     YEAR(website_sessions.created_at) AS yr,
96     MONTH(website_sessions.created_at) AS mo,
97     COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
98     COUNT(DISTINCT orders.order_id) AS orders,
99     COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conversion_rate
100 FROM website_sessions
101     LEFT JOIN orders
102         ON orders.website_session_id = website_sessions.website_session_id
103 WHERE website_sessions.created_at < '2012-11-27'
104 GROUP BY 1,2;

```

```

106 /*
107 6. For the gsearch lander test, please estimate the revenue that test earned us
108 (Hint: Look at the increase in CVR from the test (Jun 19 - Jul 28), and use
109 nonbrand sessions and revenue since then to calculate incremental value)
110 */

```

```

112 • USE mavenfuzzyfactory;

```

```

114 • SELECT
115     MIN(website_pageview_id) AS first_test_pv
116 FROM website_pageviews
117 WHERE pageview_url = '/lander-1';

```

```

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

```

```

123 • CREATE TEMPORARY TABLE first_test_pageviews
124 SELECT
125     website_pageviews.website_session_id,
126     MIN(website_pageviews.website_pageview_id) AS min_pageview_id
127 FROM website_pageviews
128     INNER JOIN website_sessions
129         ON website_sessions.website_session_id = website_pageviews.website_session_id
130         AND website_sessions.created_at < '2012-07-28' -- prescribed by the assignment
131         AND website_pageviews.website_pageview_id >= 23504 -- first page_view
132         AND utm_source = 'gsearch'
133         AND utm_campaign = 'nonbrand'
134 GROUP BY
135     website_pageviews.website_session_id;

```

```

138 • CREATE TEMPORARY TABLE nonbrand_test_sessions_w_landing_pages
139 SELECT
140     first_test_pageviews.website_session_id,
141     website_pageviews.pageview_url AS landing_page
142 FROM first_test_pageviews
143     LEFT JOIN website_pageviews
144         ON website_pageviews.website_pageview_id = first_test_pageviews.min_pageview_id
145 WHERE website_pageviews.pageview_url IN ('/home', '/lander-1');
146
147 -- SELECT * FROM nonbrand_test_sessions_w_landing_pages;
148
149 -- then we make a table to bring in orders
150 • CREATE TEMPORARY TABLE nonbrand_test_sessions_w_orders
151 SELECT
152     nonbrand_test_sessions_w_landing_pages.website_session_id,
153     nonbrand_test_sessions_w_landing_pages.landing_page,
154     orders.order_id AS order_id
155
156 FROM nonbrand_test_sessions_w_landing_pages
157 LEFT JOIN orders
158     ON orders.website_session_id = nonbrand_test_sessions_w_landing_pages.website_session_id
159 ;
160
161 • SELECT * FROM nonbrand_test_sessions_w_orders;
162
163 -- to find the difference between conversion rates
164 • SELECT
165     landing_page,
166     COUNT(DISTINCT website_session_id) AS sessions,
167     COUNT(DISTINCT order_id) AS orders,
168     COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id) AS conv_rate
169 FROM nonbrand_test_sessions_w_orders
170 GROUP BY 1;
171
172 -- .0319 for /home, vs .0406 for /lander-1
173 -- .0087 additional orders per session
174
175 -- finding the most recent pageview for gsearch nonbrand where the traffic was sent to /home
176 • SELECT
177     MAX(website_sessions.website_session_id) AS most_recent_gsearch_nonbrand_home_pageview
178 FROM website_sessions
179     LEFT JOIN website_pageviews

```

```

102         ON orders.website_session_id = website_sessions.website_session_id
103 WHERE website_sessions.created_at < '2012-11-27'
104 GROUP BY 1,2;
105
106 /*
107 6. For the gsearch lander test, please estimate the revenue that test earned us
108 (Hint: Look at the increase in CVR from the test (Jun 19 - Jul 28), and use
109 nonbrand sessions and revenue since then to calculate incremental value)
110 */
111
112 • USE mavenfuzzyfactory;
113
114 • SELECT
115     MIN(website_pageview_id) AS first_test_pv
116 FROM website_pageviews
117 WHERE pageview_url = '/lander-1';
118
119
120
121 -- for this step, we'll find the first pageview id
122
123 • CREATE TEMPORARY TABLE first_test_pageviews
124 SELECT
125     website_pageviews.website_session_id,
126     MIN(website_pageviews.website_pageview_id) AS min_pageview_id
127 FROM website_pageviews
128     INNER JOIN website_sessions
129         ON website_sessions.website_session_id = website_pageviews.website_session_id
130         AND website_sessions.created_at < '2012-07-28' -- prescribed by the assignment
131         AND website_pageviews.website_pageview_id >= 23504 -- first page_view
132         AND utm_source = 'gsearch'
133         AND utm_campaign = 'nonbrand'
134 GROUP BY
135     website_pageviews.website_session_id;
136
137 -- next, we'll bring in the landing page to each session, like last time, but restricting to home or lander-1 this time
138 • CREATE TEMPORARY TABLE nonbrand_test_sessions_w_landing_pages
139 SELECT
140     first_test_pageviews.website_session_id,
141     website_pageviews.pageview_url AS landing_page
142 FROM first_test_pageviews
143     LEFT JOIN website_pageviews

```



```

144     ON website_pageviews.website_pageview_id = first_test_pageviews.min_pageview_id
145 WHERE website_pageviews.pageview_url IN ('/home', '/lander-1');
146
147 -- SELECT * FROM nonbrand_test_sessions_w_landing_pages;
148
149 -- then we make a table to bring in orders
150 • CREATE TEMPORARY TABLE nonbrand_test_sessions_w_orders
151 SELECT
152     nonbrand_test_sessions_w_landing_pages.website_session_id,
153     nonbrand_test_sessions_w_landing_pages.landing_page,
154     orders.order_id AS order_id
155
156 FROM nonbrand_test_sessions_w_landing_pages
157 LEFT JOIN orders
158     ON orders.website_session_id = nonbrand_test_sessions_w_landing_pages.website_session_id
159 ;
160
161 • SELECT * FROM nonbrand_test_sessions_w_orders;
162
163 -- to find the difference between conversion rates
164 • SELECT
165     landing_page,
166     COUNT(DISTINCT website_session_id) AS sessions,
167     COUNT(DISTINCT order_id) AS orders,
168     COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id) AS conv_rate
169 FROM nonbrand_test_sessions_w_orders
170 GROUP BY 1;
171
172 -- .0319 for /home, vs .0406 for /lander-1
173 -- .0087 additional orders per session
174
175 -- finding the most recent pageview for gsearch nonbrand where the traffic was sent to /home
176 • SELECT
177     MAX(website_sessions.website_session_id) AS most_recent_gsearch_nonbrand_home_pageview
178 FROM website_sessions
179     LEFT JOIN website_pageviews
180         ON website_pageviews.website_session_id = website_sessions.website_session_id
181 WHERE utm_source = 'gsearch'
182     AND utm_campaign = 'nonbrand'
183     AND pageview_url = '/home'
184     AND website_sessions.created_at < '2012-11-27'
185 ;

```

```

189 • SELECT
190     COUNT(website_session_id) AS sessions_since_test
191 FROM website_sessions
192 WHERE created_at < '2012-11-27'
193     AND website_session_id > 17145 -- last /home session
194     AND utm_source = 'gsearch'
195     AND utm_campaign = 'nonbrand'
196 ;
197 -- 22,972 website sessions since the test
198
199 -- X .0087 incremental conversion = 202 incremental orders since 7/29
200     -- roughly 4 months, so roughly 50 extra orders per month. Not bad!
201
202
203
204 /*
205 7. For the landing page test you analyzed previously, it would be great to show a full conversion funnel
206 from each of the two pages to orders. You can use the same time period you analyzed last time (Jun 19 - Jul 28).
207 */
208
209 • SELECT
210     website_sessions.website_session_id,
211     website_pageviews.pageview_url,
212     -- website_pageviews.created_at AS pageview_created_at,
213     CASE WHEN pageview_url = '/home' THEN 1 ELSE 0 END AS homepage,
214     CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS custom_lander,
215     CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END AS products_page,
216     CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS mrfuzzy_page,
217     CASE WHEN pageview_url = '/cart' THEN 1 ELSE 0 END AS cart_page,
218     CASE WHEN pageview_url = '/shipping' THEN 1 ELSE 0 END AS shipping_page,
219     CASE WHEN pageview_url = '/billing' THEN 1 ELSE 0 END AS billing_page,
220     CASE WHEN pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS thankyou_page
221 FROM website_sessions
222     LEFT JOIN website_pageviews
223         ON website_sessions.website_session_id = website_pageviews.website_session_id
224 WHERE website_sessions.utm_source = 'gsearch'
225     AND website_sessions.utm_campaign = 'nonbrand'
226     AND website_sessions.created_at < '2012-07-28'
227     AND website_sessions.created_at > '2012-06-19'
228 ORDER BY
229     website_sessions.website_session_id,
230     website_pageviews.created_at;

```

```
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
    website_sessions.website_session_id,
    website_pageviews.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
    LEFT JOIN website_pageviews
        ON website_sessions.website_session_id = website_pageviews.website_session_id
WHERE website_sessions.utm_source = 'gsearch'
    AND website_sessions.utm_campaign = 'nonbrand'
    AND website_sessions.created_at < '2012-07-28'
        AND website_sessions.created_at > '2012-06-19'
ORDER BY
    website_sessions.website_session_id,
    website_pageviews.created_at
) AS pageview_level

GROUP BY
    website_session_id
;
```