eCommerce Database Analysis with SQL

1. Project Background

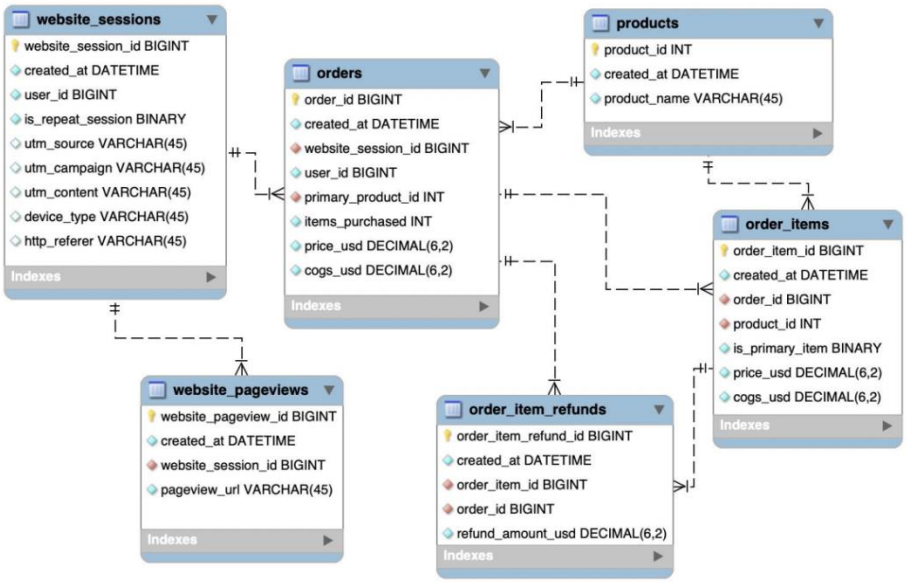
You've been hired as an eCommerce Database Analyst for Maven Fuzzy Factory, an online retailer which has just launched its first product.

Maven Fuzzy Factory has been live for ~8 months, and your CEO is due to present company performance metrics to the board next week.

You will extract and analyze website traffic and performance data from the Maven Fuzzy Factory database to quantify the company's growth and tell the story of how you have been able to generate that growth.

2. Data Preparation

2.1. Entity Relationship Diagram (ERD)



2.2. Tables

a. Orders

Records consist of customers' orders with order id, time when the order is created, website session id, user id, product id of item ordered, number of items purchased, the price of the product (revenue) and COGS (cost of goods sold) in USD. There are 32,313 rows in this table.

order_id	created_at	website_session_id	user_id	primary_product_id	items_purchased	price_usd	cogs_usd
1	2012-03-19 10:42:46	20	20	1	1	49.99	19.49
2	2012-03-19 19:27:37	104	104	1	1	49.99	19.49
3	2012-03-20 06:44:45	147	147	1	1	49.99	19.49
4	2012-03-20 09:41:45	160	160	1	1	49.99	19.49
5	2012-03-20 11:28:15	177	177	1	1	49.99	19.49

b. Products

Records consist of customers' orders with order id, time when the order is created, website session id, user id, product id of item ordered, number of items purchased, the price of the product (revenue), and cogs (cost of goods sold) in USD. There are 4 rows in this table.

product_id	created_at	product_name
1	2012-03-19 08:00:00	The Original Mr. Fuzzy
2	2013-01-06 13:00:00	The Forever Love Bear
3	2013-12-12 09:00:00	The Birthday Sugar Panda
4	2014-02-05 10:00:00	The Hudson River Mini bear

c. Website_sessions

Records consist of each website session. This table shows where the traffic is coming from. There are 472,871 rows in this table.

website_session_id	created_at	user_id	is_repeat_session	utm_source	utm_campaign	utm_content	device_type	http_referer
1	2012-03-19 08:04:16	1	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com
2	2012-03-19 08:16:49	2	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
3	2012-03-19 08:26:55	3	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
4	2012-03-19 08:37:33	4	0	gsearch	nonbrand	g_ad_1	desktop	https://www.gsearch.com
5	2012-03-19 09:00:55	5	0	gsearch	nonbrand	g_ad_1	mobile	https://www.gsearch.com

d. Website_pageviews

The table that shows website pageviews and url of each pageview. There are 1,188,124 rows in this table.

website_pageview_id	created_at	website_session_id	pageview_url
1	2012-03-19 08:04:16	1	/home
2	2012-03-19 08:16:49	2	/home
3	2012-03-19 08:26:55	3	/home
4	2012-03-19 08:37:33	4	/home
5	2012-03-19 09:00:55	5	/home

e. Order_items

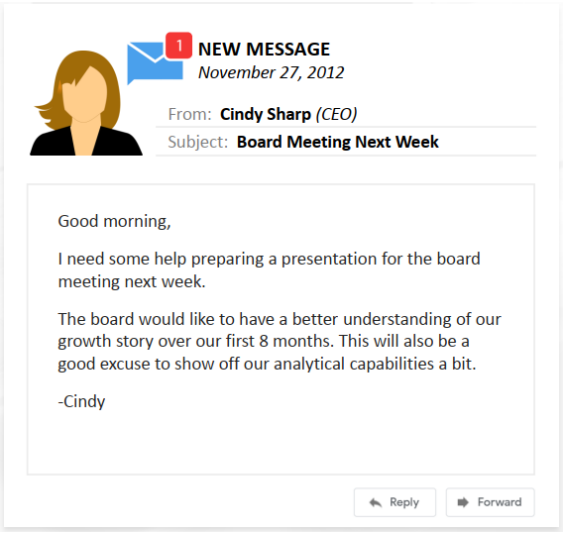
order_item_id	created_at	order_id	product_id	is_primary_item	price_usd	cogs_usd
1	2012-03-19 10:42:46	1	1	1	49.99	19.49
2	2012-03-19 19:27:37	2	1	1	49.99	19.49
3	2012-03-20 06:44:45	3	1	1	49.99	19.49
4	2012-03-20 09:41:45	4	1	1	49.99	19.49
5	2012-03-20 11:28:15	5	1	1	49.99	19.49

f. Order_item_refunds

order_item_refund_id	created_at	order_item_id	order_id	refund_amount_usd
1	2012-04-06 11:32:43	57	57	49.99
2	2012-04-13 01:09:43	74	74	49.99
3	2012-04-15 07:03:48	71	71	49.99
4	2012-04-17 20:00:37	118	118	49.99
5	2012-04-22 20:53:49	116	116	49.99

3. Project Goal

Cindy Sharp, the Maven Fuzzy Factory CEO just sent you an email. She needs your help to prepare a presentation for the board meeting.



3.1. Objectives

- Tell the story of the company's growth, using trended performance data.
- Use the database to explain some of the details around the company's growth story.
- Analyse current performance and use the data available to assess upcoming opportunities.

3.2. Problem Questions

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?
2. Next, it would be great to see a similar 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.
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.

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?
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?
6. Do a comparison for */home* and */lander-1* landing page test. 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)
7. For the landing page test that you analysed previously, it would be great to shows a full conversion funnel from each of the two pages to orders. You can use the same period you analysed last time (Jun 19 - Jul 28).
8. I'd love for you to quantify the impact of our billing test (*billing* and *billing-2*), as well. Please analyse the lift generated from the test (Sep 10 - Nov 10), in terms of revenue per billing page sessions, and then pull the number of billing page sessions for the past month to understand monthly impact.

4. Analysis and Visualization

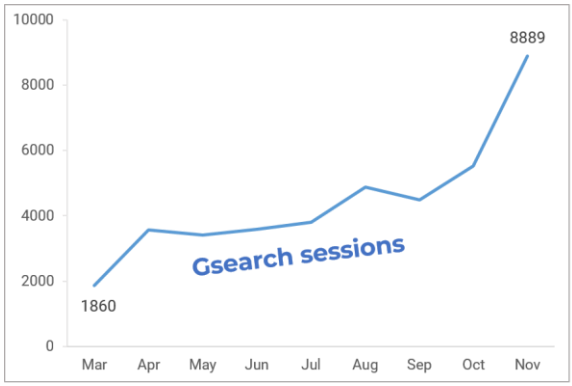
Q1. Gsearch seems to be the biggest driver of our busienss. Could you pull monthly trends for gsearch sessions and orders so that we can showcase the growth there?

```
1. SELECT
2.    EXTRACT(YEAR_MONTH FROM website_sessions.created_at) AS yearmonth,
3.    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
4.    COUNT(DISTINCT order_id) AS orders,
5.    ROUND(COUNT(DISTINCT order_id) /
6.      COUNT(DISTINCT website_sessions.website_session_id) * 100.0, 2) AS
   conversion_rate
7. FROM website_sessions
8.    LEFT JOIN orders
9.    ON website_sessions.website_session_id = orders.website_session_id
10.WHERE utm_source = 'gsearch'
11.   AND website_sessions.created_at < '2012-11-27'
12.GROUP BY 1;
```

Result:

yearmonth	sessions	orders	conversion_rate
201203	1860	60	3.23
201204	3574	92	2.57
201205	3410	97	2.84
201206	3578	121	3.38
201207	3811	145	3.80
201208	4877	184	3.77
201209	4491	188	4.19
201210	5534	234	4.23
201211	8889	373	4.20

Insight:



Gsearch traffic shows steady growth of sessions and orders.

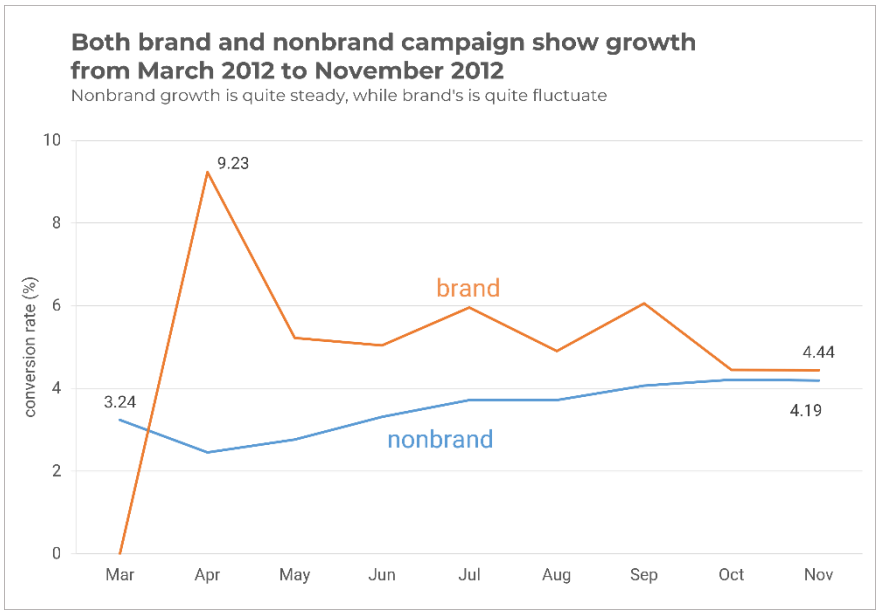
Q2. Next, it would be great to see a similar trend for gsearch, but this time splitting out nonbrand and brand campaigns separately. I am wondering if brand is picking up at all. If so, this is a good story to tell.

```
1.  SELECT
2.    EXTRACT(YEAR_MONTH FROM website_sessions.created_at) AS yearmonth,
3.    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN
    website_sessions.website_session_id ELSE NULL END) AS nonbrand_sessions,
4.    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN orders.order_id
    ELSE NULL END) AS nonbrand_orders,
5.    ROUND(COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN
    orders.order_id ELSE NULL END) /
6.      COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN
    website_sessions.website_session_id ELSE NULL END) * 100.0, 2) AS
    nonbrand_cvr,
7.    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN
    website_sessions.website_session_id ELSE NULL END) AS brand_sessions,
8.    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN orders.order_id
    ELSE NULL END) AS brand_orders,
9.    ROUND(COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN
    orders.order_id ELSE NULL END) /
10.     COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN
    website_sessions.website_session_id ELSE NULL END) * 100.0, 2) AS
    brand_cvr
11. FROM website_sessions
12.   LEFT JOIN orders
13.     ON website_sessions.website_session_id = orders.website_session_id
14. WHERE utm_source = 'gsearch'
15.   AND website_sessions.created_at < '2012-11-27'
16. GROUP BY 1;
```

Result:

yearmonth	nonbrand_sessions	nonbrand_orders	nonbrand_cvr	brand_sessions	brand_orders	brand_cvr
201203	1852	60	3.24	8	0	0.00
201204	3509	86	2.45	65	6	9.23
201205	3295	91	2.76	115	6	5.22
201206	3439	114	3.31	139	7	5.04
201207	3660	136	3.72	151	9	5.96
201208	4673	174	3.72	204	10	4.90
201209	4227	172	4.07	264	16	6.06
201210	5197	219	4.21	337	15	4.45
201211	8506	356	4.19	383	17	4.44

Insight:



For *brand* campaign during April, the conversion rate is very high at 9.23%, though the number of sessions and orders are still considered low compared to *nonbrand* campaign. The *brand* sessions and orders do increase steadily every month, and while still lower than *nonbrand*, in November its conversion rate shows higher number than *nonbrand* (4.44% vs 4.19% for *nonbrand* and *brand*, respectively).

Q3. 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.

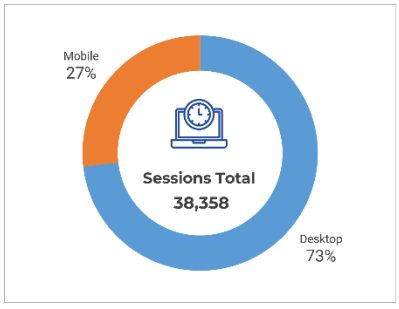
```
1.  SELECT
2.    EXTRACT(YEAR_MONTH FROM website_sessions.created_at) AS yearmonth,
3.    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN
    website_sessions.website_session_id ELSE NULL END) AS desktop_sessions,
4.    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN orders.order_id
    ELSE NULL END) AS desktop_orders,
5.      ROUND(COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN
    orders.order_id ELSE NULL END) /
```

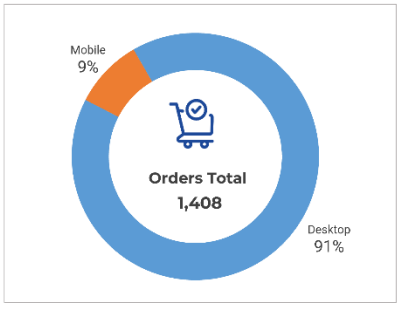
```
6.    COUNT(DISTINCT CASE WHEN device_type = 'desktop' THEN
    website_sessions.website_session_id ELSE NULL END) * 100.0, 2) AS
    desktop_cvr,
7.    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN
    website_sessions.website_session_id ELSE NULL END) AS mobile_sessions,
8.    COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN orders.order_id
    ELSE NULL END) AS mobile_orders,
9.      ROUND(COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN
    orders.order_id ELSE NULL END) /
10.   COUNT(DISTINCT CASE WHEN device_type = 'mobile' THEN
    website_sessions.website_session_id ELSE NULL END) * 100.0, 2) AS
    mobile_cvr
11. FROM website_sessions
12.   LEFT JOIN orders
13.     ON website_sessions.website_session_id = orders.website_session_id
14. WHERE utm_source = 'gsearch'
15.   AND utm_campaign = 'nonbrand'
16.   AND website_sessions.created_at < '2012-11-27'
17. GROUP BY 1;
```

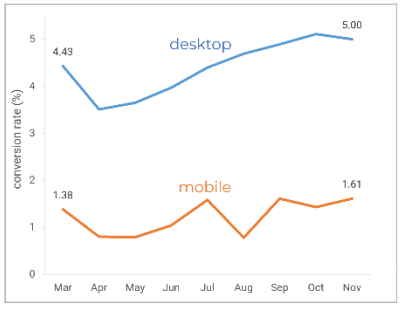
Result:

yearmonth	desktop_sessions	desktop_orders	desktop_cvr	mobile_sessions	mobile_orders	mobile_cvr
201203	1128	50	4.43	724	10	1.38
201204	2139	75	3.51	1370	11	0.80
201205	2276	83	3.65	1019	8	0.79
201206	2673	106	3.97	766	8	1.04
201207	2774	122	4.40	886	14	1.58
201208	3515	165	4.69	1158	9	0.78
201209	3171	155	4.89	1056	17	1.61
201210	3934	201	5.11	1263	18	1.43
201211	6457	323	5.00	2049	33	1.61

Insight:







Majority of traffic sources are coming from users on desktop. Both desktop and mobile shows increased conversion rate from March to November 2012. Investigate why there are less session and orders from users who access through mobile, look into the the mobile webpages user interface and experience.

Q4. 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?

First, find the various utm sources and referers to see the traffic we're getting.

```
1.  SELECT DISTINCT
2.    utm_source,
3.    utm_campaign,
4.    http_referer
5.  FROM website_sessions
6.  WHERE website_sessions.created_at < '2012-11-27';
```

Result:

utm_source	utm_campaign	http_referer
gsearch	nonbrand	https://www.gsearch.com
NULL	NULL	NULL
gsearch	brand	https://www.gsearch.com
NULL	NULL	https://www.gsearch.com
bsearch	brand	https://www.bsearch.com
NULL	NULL	https://www.bsearch.com
bsearch	nonbrand	https://www.bsearch.com

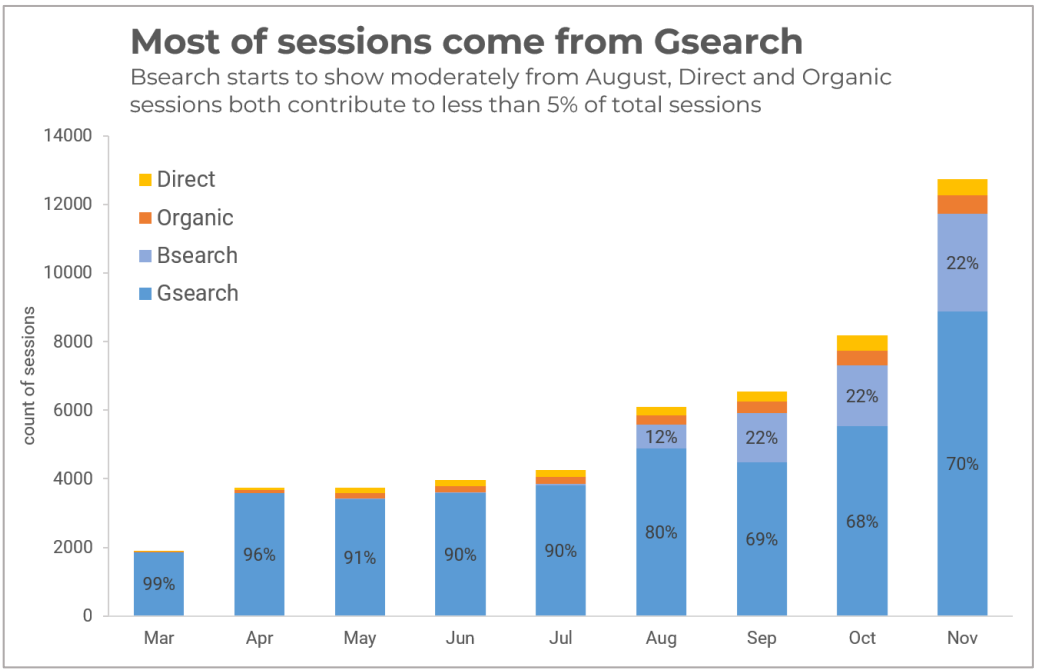
- If *utm_source* and *utm_campaign* IS NULL and *http_referer* IS NOT NULL, it means the sessions come from organic search sessions.
- If *utm_source* and *utm_campaign* IS NULL and *http_referer* IS NULL, it means the sessions come directly from the web / users directly type the website link.

```
1. SELECT
2.   EXTRACT(YEAR_MONTH FROM website_sessions.created_at) AS yearmonth,
3.   COUNT(website_sessions.website_session_id) AS sessions,
4.   COUNT(DISTINCT CASE WHEN utm_source = 'gsearch' AND http_referer IS NOT
   NULL THEN website_sessions.website_session_id ELSE NULL END) AS
   gsearch_paid_sessions,
5.   COUNT(DISTINCT CASE WHEN utm_source = 'bsearch' AND http_referer IS NOT
   NULL THEN website_sessions.website_session_id ELSE NULL END) AS
   bsearch_paid_sessions,
6.   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,
7.   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_sessions
8. FROM website_sessions
9.   LEFT JOIN orders
10.     ON website_sessions.website_session_id = orders.website_session_id
11. WHERE website_sessions.created_at < '2012-11-27'
12. GROUP BY 1;
```

Result:

yearmonth	sessions	gsearch_paid_sessions	bsearch_paid_sessions	organic_search_sessions	direct_type_sessions
201203	1879	1860	2	8	9
201204	3734	3574	11	78	71
201205	3736	3410	25	150	151
201206	3963	3578	25	190	170
201207	4249	3811	44	207	187
201208	6097	4877	705	265	250
201209	6546	4491	1439	331	285
201210	8183	5534	1781	428	440
201211	12750	8889	2840	536	485

Insight:



Number of sessions keep growing every month. Large portion of sessions come from gsearch, starting at 99% at March though it starts to decreased and contribute to 70% of total sessions in November. Bsearch traffic starts to grow in August, contributing to 12% of total and reach its highest in November at 22% of total sessions.

Q5. 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?

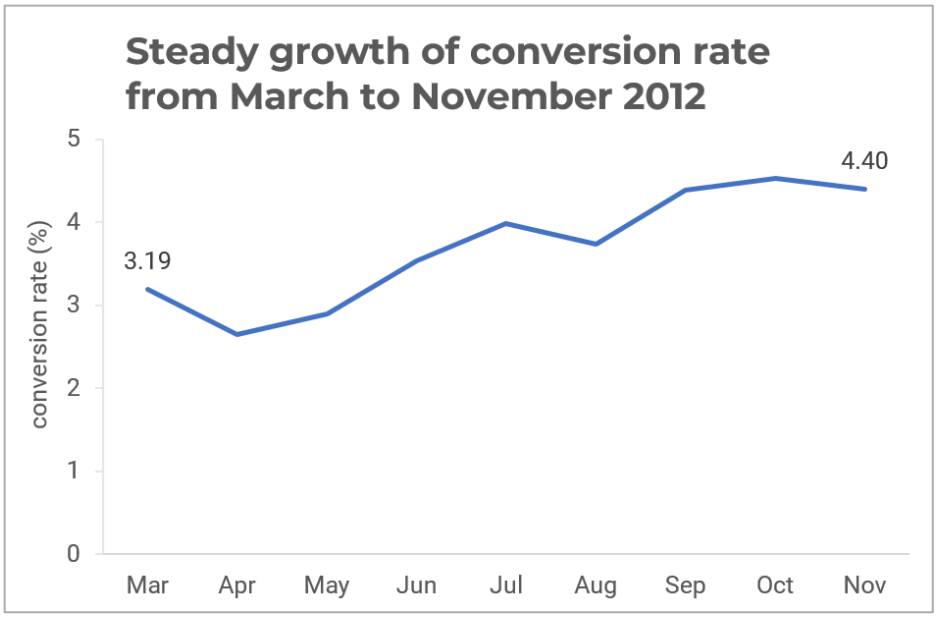
```
1. SELECT
2.    EXTRACT(YEAR_MONTH FROM website_sessions.created_at) AS yearmonth,
3.    COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
4.    COUNT(DISTINCT order_id) AS orders,
5.    ROUND(COUNT(DISTINCT order_id) / COUNT(DISTINCT
   website_sessions.website_session_id)*100.0, 2) AS conversion_rate
6. FROM website_sessions
7.    LEFT JOIN orders
```

```
8.      ON website_sessions.website_session_id = orders.website_session_id
9.  WHERE website_sessions.created_at < '2012-11-27'
10. GROUP BY 1;
```

Result:

yearmonth	sessions	orders	conversion_rate
201203	1879	60	3.19
201204	3734	99	2.65
201205	3736	108	2.89
201206	3963	140	3.53
201207	4249	169	3.98
201208	6097	228	3.74
201209	6546	287	4.38
201210	8183	371	4.53
201211	12750	561	4.40

Insight:



The conversion rate has been growing steadily, from 3.19% in March to 4.40% in November.

Q6. 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)

First, we search the first website pageview id for */lander-1* url.

```
1. SELECT
2.   MIN(website_pageview_id) AS first_test_pv
3. FROM website_pageviews
4. WHERE pageview_url = '/lander-1';
```

Result:

first_test_pv
23504

The first website pageview id for */lander-1* test page is 23504.

```
1. SELECT
2.   website_pageviews.pageview_url AS landing_page,
3.   COUNT(DISTINCT website_sessions.website_session_id) AS sessions,
4.   COUNT(DISTINCT orders.order_id) AS orders,
5.   ROUND(COUNT(DISTINCT orders.order_id)/
6.     COUNT(DISTINCT website_sessions.website_session_id) * 100.0,2) AS
   conversion_rate
7. FROM website_sessions
8. INNER JOIN website_pageviews
9.   ON website_sessions.website_session_id =
   website_pageviews.website_session_id
10. LEFT JOIN orders
11.   ON website_sessions.website_session_id = orders.website_session_id
12. WHERE website_pageviews.website_pageview_id >= 23504
13.   AND website_sessions.created_at < '2012-07-28'
14.   AND website_sessions.utm_source = 'gsearch'
15.   AND website_sessions.utm_campaign = 'nonbrand'
16.   AND website_pageviews.pageview_url IN ('/home', '/lander-1')
17. GROUP BY website_pageviews.pageview_url;
```

Result:

landing_page	sessions	orders	conversion_rate
/home	2261	72	3.18
/lander-1	2316	94	4.06

- Homepage lander conversion rate is 3.18%, while new test lander page conversion rate is 4.06%. The conversion rate is increased by 0.88%.
- To calculate estimate revenue generated by new test lander page, first we find the last time */home* page appeared, then we count the total sessions since that.

```
1. SELECT
2.   MAX(website_sessions.website_session_id) AS
   most_recent_gsearch_nonbrand_home_pageview
3. FROM website_sessions
4.   LEFT JOIN website_pageviews
5.   ON website_sessions.website_session_id =
   website_pageviews.website_session_id
6. WHERE utm_source = 'gsearch'
7.   AND utm_campaign = 'nonbrand'
8.   AND pageview_url = '/home' -- Home landing page
9.   AND website_sessions.created_at < '2012-11-27';
```

Result:

most_recent_gsearch_nonbrand_home_pageview
17145

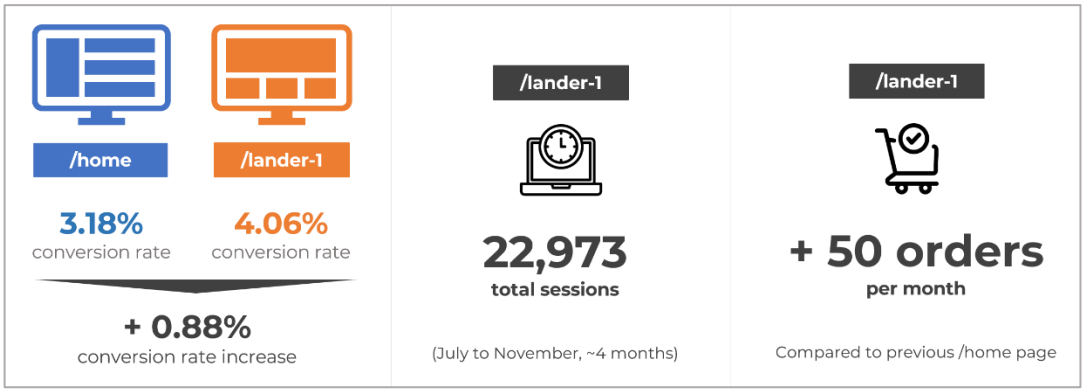
- Max website_session_id for */home* is 17145.
- After this session, there are no more /home landing page, and all landing page has been replaced with */lander-1*.

```
1.  SELECT
2.    COUNT(website_session_id) AS sessions_since_test
3.  FROM website_sessions
4.  WHERE
5.    created_at < '2012-11-27'
6.    AND website_session_id >= 17145 -- last home session
7.    AND utm_source = 'gsearch'
8.    AND utm_campaign = 'nonbrand';
```

sessions_since_test
22973

Calculate Average Revenue:

- Conversion rate difference: 0.88%
- Total of sessions using */lander-1* = 22,973
- 22,973 x 0.88% = estimated at least 202 incremental orders since July 29 using */lander-1* page for roughly 4 months
- 202/4 = 50 additional orders per month. Awesome!!



Q7. For the landing page test you analyzed previously, it would be great to shows a full conversion funnel from each of the two pages to orders. You can use the same time period you analyzed last time (Jun 19 - Jul 28)

```sql
SELECT
  MIN(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url = '/lander-1';

-- First test lander-1 pageviews is 23504

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 home_page,
  CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS lander1_page,
  CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END AS product_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_pageview_id >= 23504
  AND website_pageviews.created_at < '2012-07-28'
  AND website_pageviews.pageview_url IN ('/home', '/lander-1', '/products', '/the-original-mr-fuzzy', '/cart', '/shipping', '/billing', '/thank-you-for-your-order')
ORDER BY
  website_sessions.website_session_id,
  website_pageviews.created_at
;

-- next we will put the previous query inside a subquery (similar to temporary tables)
-- we will group by website_session_id, and take the MAX() of each of the flags
-- this MAX() becomes a made it flag for that session, to show the session made it there

CREATE TEMPORARY TABLE session_level_made_it_flags
SELECT
  website_session_id,
  MAX(homepage) AS saw_homepage,
  MAX(custom_lander) AS saw_custom_lander,
  MAX(product_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,
```

```sql
  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 product_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_pageviews.created_at < '2012-07-28'
  AND website_pageviews.created_at > '2012-06-19'
ORDER BY
  website_sessions.website_session_id,
  website_pageviews.created_at
) AS pageview_level
GROUP BY 1;

SELECT *
FROM session_level_made_it_flags;

-- then this will produce the final output (part 1)

SELECT
  CASE
    WHEN saw_homepage = 1 THEN 'saw_homepage'
    WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
    ELSE 'uh oh... check logic'
    END AS segment,
  COUNT(DISTINCT website_session_id) AS sessions,
  COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id
ELSE NULL END) AS to_products,
  COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id
ELSE NULL END) AS to_mrfuzzy,
  COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE
NULL END) AS to_cart,
  COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN website_session_id
ELSE NULL END) AS to_shipping,
```

```sql
  COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN website_session_id
ELSE NULL END) AS to_billing,
  COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN website_session_id
ELSE NULL END) AS to_thankyou
FROM
  session_level_made_it_flags
GROUP BY 1;

-- then this is the final output part 2, click rates or conversion rates
-- click rates or conversion rates is percentage of click rate from
certain page divided by total sessions

SELECT
  CASE
    WHEN saw_homepage = 1 THEN 'saw_homepage'
    WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
    ELSE 'uh oh... check logic'
    END AS segment,
  ROUND(COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN
website_session_id ELSE NULL END) /
    COUNT(DISTINCT website_session_id) * 100.0, 2) AS products_click_rt,
  ROUND(COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN
website_session_id ELSE NULL END) /
    COUNT(DISTINCT website_session_id) * 100.0, 2) AS mrfuzzy_click_rt,
  ROUND(COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN
website_session_id ELSE NULL END) /
    COUNT(DISTINCT website_session_id) * 100.0, 2) AS cart_click_rt,
  ROUND(COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN
website_session_id ELSE NULL END) /
    COUNT(DISTINCT website_session_id) * 100.0, 2) AS shipping_click_rt,
  ROUND(COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN
website_session_id ELSE NULL END) /
    COUNT(DISTINCT website_session_id) * 100.0, 2) AS billing_click_rt,
  ROUND(COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN
website_session_id ELSE NULL END) /
    COUNT(DISTINCT website_session_id) * 100.0, 2) AS thankyou_click_rt
FROM
  session_level_made_it_flags
GROUP BY 1;
```

Result:

- Output 1: Sessions Funnel

segment	sessions	to_products	to_mrfuzzy	to_cart	to_shipping	to_billing	to_thankyou
saw_custom_lander	2316	1083	772	348	231	197	94
saw_homepage	2261	942	684	296	200	168	72

- Output 2: Click-rates Funnel

segment	products_click_rt	mrfuzzy_click_rt	cart_click_rt	shipping_click_rt	billing_click_rt	thankyou_click_rt
saw_custom_lander	46.76	33.33	15.03	9.97	8.51	4.06
saw_homepage	41.66	30.25	13.09	8.85	7.43	3.18

Q8. 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 sessions, and then pull the number of billing page sessions for the past month to understand monthly impact.

```
1.  SELECT
2.    billing_version_seen,
3.    COUNT(DISTINCT website_session_id) AS sessions,
4.    ROUND(SUM(price_usd) / COUNT(DISTINCT website_session_id), 2) AS
    revenue_per_session
5.  FROM
6.  (
7.  SELECT
8.    website_pageviews.website_session_id,
9.    website_pageviews.pageview_url AS billing_version_seen,
10.   orders.order_id,
11.   orders.price_usd
12. FROM website_pageviews
13.   LEFT JOIN orders
14.     ON website_pageviews.website_session_id = orders.website_session_id
15. WHERE
16.   website_pageviews.created_at BETWEEN '2012-09-10' and '2012-11-10'
17.   AND website_pageviews.pageview_url IN ('/billing', '/billing-2')
18. ) AS billing_pageviews_and_order_data
19. GROUP BY 1;
```

Result:

billing_version_seen	sessions	revenue_per_session
/billing	657	22.83
/billing-2	654	31.34

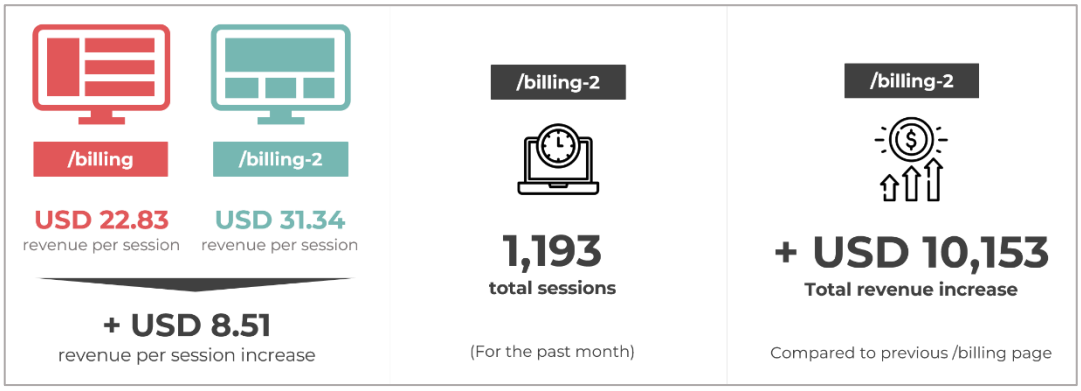
- */billing* page generates 657 sessions, with average USD 22,83 revenue per session
- */billing-2* page generates 654 sessions, with average USD 31,34 revenue per session
- **INCREASE: USD 8.51 per session**

```
1.  SELECT
2.    COUNT(website_session_id) AS billing_sessions_past_month
3.  FROM website_pageviews
4.  WHERE created_at BETWEEN '2012-10-27' AND '2012-11-27'
5.    AND pageview_url IN ('/billing', '/billing-2');
```

Result:

billing_sessions_past_month
1193

- */billing* page USD 22,83 revenue per session and new */billing-2* page generates USD 31,34 revenue per session. The lift is **USD 8.51 per session**.
- Over the past month there has been 1,193 sessions. The new page has generated **USD 10,153 increase in revenue**.



5. Insight and Recommendation

5.1. Insight:

1. The website performance has seen improvements over the course of the first 8 months. The conversion rate starts at 3.19% in March and reached 4.40% in November.
2. Most of our traffic comes from users who access from desktop, almost 3/4 of traffic comes from desktop, while the rest comes from mobile.
3. In March, 99% of our traffic comes from gsearch. From August to November, our traffic sources are more diverse; 70% of it comes from gsearch, 22% comes from bsearch, and 8% comes from direct and organic.
4. The new */lander-1* test shows better conversion rate compared to original */home* page. The conversion rose from 3.18% to 4.06%, adding an increase of 0.88%. The */lander-1* page generated additional 50 orders per month and shows better click rates funnel.
5. The new */billing-2* test page also shows better result than the previous */billing* page, which brought additional USD 8.51 revenue per session. For the past month, there has been 1,193 session total, and the new billing page brought a total of USD 10,153 increase in revenue.

5.2. Recommendation:

1. With most of our users coming from desktop, we can focus our campaign and budget to desktop users. Additionally, evaluate the mobile webpage and find why the traffic is low, then create better user interface and experience for mobile users.
2. For paid marketing campaigns, most of the sessions come from gsearch than bsearch. We can focus our budget and campaign to gsearch for higher sessions reach in the future.