# Analyzing Website Performance

## 1 – Identifying Top Website Pages & Tope Entry Pages

Use Website Pageviews data to find the most-viewed pages by your users, and the most common entry pages to your website, and to evaluate how those pages perform for your business objectives.

Assignment 1.1: Help Website Manager find the Top Website Pages ranked by session volume.

```sql
-- Most viewed Pages
SELECT
    pageview_url,
    COUNT(DISTINCT website_pageview_id) AS pvs
FROM website_pageviews
WHERE created_at < '2012-06-09' -- date of assignment
GROUP BY pageview_url
ORDER BY pvs DESC;
```

| pageview_url | pvs |
|---|---|
| /home | 10398 |
| /products | 4238 |
| /the-original-mr-fuzzy | 3036 |
| /cart | 1305 |
| /shipping | 869 |
| /billing | 716 |
| /thank-you-for-your-order | 306 |

The /homepage generates the highest volume of page viewed during the time period, followed by are the multi-product showcase page (/product), and the specific product Mr. Fuzzy page. This indicates that these pages should be focused on while working on website performance improvement.

Assignment 1.2: Pull a list of Top Entry Pages where customers land on your site for the first time.

```sql
-- Finding Top Entry Pages
-- Step 1:find the first pageview for each session (the 1st pageview a customer saw on a given session)
-- Step 2:find the URL the customer saw on that first pageview
WITH first_pageview -- first_pv_per_session
AS (
    SELECT
        website_session_id,
        MIN(website_pageview_id) AS min_pageview_id -- auto incrementing
    FROM website_pageviews
    WHERE created_at <'2012-06-12' -- date of assignment
    GROUP BY 1) -- the 1st pageview for each session

   -- take that pageview_id join back to the table and find that URL
SELECT
    t2.pageview_url AS landing_page,-- aka "entry page"
    COUNT(DISTINCT t1.website_session_id) AS sessions_hitting_this_lander
FROM first_pageview t1
    LEFT JOIN website_pageviews t2
      ON t1.min_pageview_id = t2.website_pageview_id
GROUP BY landing_page; -- count of sessions by landing page URL
```

| landing_page | sessions_hitting_this_lander |
|---|---|
| /home | 10711 |

At this time in the life of the business, our traffic comes in through the homepage. Again, this is clearly where to focus on making any improvements so as to ensure best initial experience for all customers.

# 2 – Analyzing Landing Page Performance & Testing

Measure the performance of key landing pages and optimization testing results to improve performance.

## Assignment 2.1: Calculating Bounce Rates & Analyzing Landing Page Tests

```sql
-- Step 0: Find out when the new page (/Lander1) launched - fair comparison
SELECT
    MIN(created_at) AS first_created_at,
    MIN(website_pageview_id) AS first_pageview_id
FROM website_pageviews
WHERE pageview_url ='/lander-1'
    AND created_at IS NOT NULL;
```

| first_created_at | first_pageview_id |
|---|---|
| 2012-06-19 01:35:54 | 23504 |

```sql
WITH t1 AS  -- Step 1: Find the 1st website_pageview_id (MIN) for each relevant session
(
    SELECT
        wp.website_session_id AS website_session_id,
        MIN(wp.website_pageview_id) AS min_pageview_id
    FROM website_pageviews wp JOIN website_sessions ws
        ON wp.website_session_id = ws.website_session_id
        AND ws.created_at < '2012-07-28' -- date of assignment
        AND wp.website_pageview_id > 23504 -- 1st pv_id for /lander1
        AND ws.utm_source = 'gsearch'
        AND ws.utm_campaign = 'nonbrand'
GROUP BY 1),

t2 AS  -- Step 2: Identify the landing page (URL) of each session
(
    SELECT
        t1.website_session_id AS session_id,
        wp.pageview_url AS landing_page
    FROM t1 LEFT JOIN website_pageviews wp
      ON t1.min_pageview_id = wp.website_pageview_id
    WHERE wp.pageview_url IN ('/home', '/lander-1')),

t3 AS  -- Step 3: Count (total) pageviews for each session to identify "Bounces"
(
    SELECT
        t2.session_id AS bounced_session_id,
        t2.landing_page,
        COUNT(wp.website_pageview_id) AS count_of_pages_viewed
    FROM t2 LEFT JOIN website_pageviews wp
      ON t2.session_id = wp.website_session_id
    GROUP BY 1,2
    HAVING count_of_pages_viewed = 1)-- limit to those have a count equals to 1

SELECT  -- Step 4: Summary of total sessions and bounced sessions by LP
    t2.landing_page, -- our Dimension
    COUNT(DISTINCT t2.session_id) AS total_sessions,
    COUNT(DISTINCT t3.bounced_session_id) AS bounced_sessions,
    COUNT(DISTINCT t3.bounced_session_id)/COUNT(DISTINCT t2.session_id) AS bounce_rate
FROM t2 LEFT JOIN t3
    ON t2.session_id = t3.bounced_session_id
GROUP BY t2.landing_page;
```

| | landing_page | total_sessions | bounced_sessions | bounce_rate |
|---|---|---|---|---|
| ▶ | /home | 2260 | 1319 | 0.5836 |
| | /lander-1 | 2313 | 1231 | 0.5322 |

The new lander (/lander-1), which is the custom landing page the Website Manager set up has a bounce rate of **53%** versus the /homepage for the same traffic was at **58%**. Given performance improvement that **the custom landing page is causing fewer customers to bounce**, future nonbrand paid traffic would better be routed to the new page.

**Assignment 2.2: Analyzing Sales Lift** - For the **Gsearch lander test** (Assignment 2.1), estimate the sales lift that test generated.

```
-- Step 1: Find the MIN pageview_id where that test started
SELECT
    MIN(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url ='/lander-1'; -- 23504

WITH t1 AS  -- Step 2: Find the first test pageviews
(
    SELECT
      p.website_session_id AS session_id,
      MIN(p.website_pageview_id) AS min_pageview_id
    FROM website_pageviews p JOIN website_sessions s
      ON p.website_session_id = s.website_session_id
      AND s.created_at < '2012-07-28' -- prescribed by the assignment
      AND p.website_pageview_id >= 23504 -- 1st page_view where test was live
      AND utm_source = 'gsearch'
      AND utm_campaign = 'nonbrand'
    GROUP BY 1), -- session-level table (1st pv id of each session)

t2 AS  -- Step 3: Find the LP (URL) for each session, restricting to /home or /Lander-1
(
    SELECT
      session_id AS session_id,
      w.pageview_url AS landing_page
    FROM t1 LEFT JOIN website_pageviews w
      ON t1.min_pageview_id = w.website_pageview_id
      WHERE w.pageview_url IN ('/home', '/lander-1')),

t3 AS -- Step 4: Bring in Orders:
(
    SELECT t2.session_id, t2.landing_page, o.order_id
    FROM t2 LEFT JOIN orders o ON t2.session_id = o.website_session_id)

SELECT  -- Step 5: Find the difference in Conversion Rates
    landing_page,
    COUNT(DISTINCT session_id) AS sessions,
    COUNT(DISTINCT order_id) AS orders,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT session_id) AS conv_rate -- session to order cvr
FROM t3 GROUP BY 1;
```

| | landing_page | sessions | orders | conv_rate |
|---|---|---|---|---|
| ▶ | /home | 2260 | 72 | 0.0319 |
| | /lander-1 | 2314 | 94 | 0.0406 |

The lander-1 performs fairly better than the homepage as it converts 0.87% additional orders per sessions.

```sql
-- Step 6: Find the most reent pageview for gsearch nonbrand where the traffic was sent to /home
SELECT MAX(s.website_session_id) AS most_recent_gsearch_nonbrand_home_pageview
FROM website_sessions s LEFT JOIN website_pageviews w
    ON s.website_session_id = w.website_session_id
WHERE s.utm_source = 'gsearch'
    AND s.utm_campaign ='nonbrand'
    AND w.pageview_url ='/home'
    AND s.created_at < '2012-11-27'; -- date of assignment
-- max website_session_id = 17145
-- since then all of the traffic has been rerouted to lander-1

-- Step 7: COUNT total #session since that test
SELECT COUNT(website_session_id) AS sessions_since_test
FROM website_sessions
    WHERE created_at <'2012-11-27'
    AND website_session_id > 17145
    AND utm_source = 'gsearch'
    AND utm_campaign = 'nonbrand';
-- 22,972 website sessions since the test
```

| | sessions_since_test |
|---|---|
| ▶ | 22972 |

Revenue Lift = 0.87% incremental conversion * 22,972 website sessions = 200 incremental orders:

Roughly 4 months since 2012-07-28 to 2012-11-27 – roughly 50 extra orders per month

# 3 – Analyzing & Testing Conversion Funnels

Understand and optimize each step of user's experience on their journey toward purchasing products.

Identify how many users continue onto each step in the conversion flow and how many drop off, and then optimize the critical pain points with high abandonment rates in order to convert and sell better.

## Assignment 3.1: Building & Analyzing Conversion Funnels

```sql
WITH t1 AS -- Step 1: Identify each PV as the specific funnel step
(
    SELECT
        s.website_session_id AS session_id,
        w.pageview_url AS pageview_url,
        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 shiping_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 s LEFT JOIN website_pageviews w
      ON s.website_session_id = w.website_session_id
    WHERE s.utm_source = 'gsearch'  -- specified in the assignment
        AND s.utm_campaign ='nonbrand'  -- specified in the assignment
        AND s.created_at > '2012-08-05' -- specified in the assignment
        AND s.created_at < '2012-09-05' -- date of assignment
    ORDER BY s.website_session_id, w.created_at),  -- pageview_level table

t2 AS -- Step 2: Create the session-level conversion funnel view
(
    SELECT
        session_id,
        MAX(products_page) AS product_made_it,
        MAX(mrfuzzy_page) AS mrfuzzy_made_it,
        MAX(cart_page) AS cart_made_it,
        MAX(shiping_page) AS shipping_made_it,
        MAX(billing_page) AS billing_made_it,
        MAX(thankyou_page) AS thankyou_made_it
    FROM t1
    GROUP BY 1)  -- Session-Level Summary to see how far each customer made it in the funnel
```

```sql
SELECT -- Step 3: Aggregate-Level Analysis to assess funnel performance (CTRs) across all sessions
    COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN session_id ELSE NULL END)
    /COUNT(DISTINCT session_id)  AS lander_click_rt,
    COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN session_id ELSE NULL END)
    /COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN session_id ELSE NULL END) AS products_click_rt,
    COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN session_id ELSE NULL END)
    /COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN session_id ELSE NULL END) AS mrfuzzy_click_rt,
    COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN session_id ELSE NULL END)
    /COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN session_id ELSE NULL END) AS cart_click_rt,
    COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN session_id ELSE NULL END)
    /COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN session_id ELSE NULL END) AS shipping_click_rt,
    COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN session_id ELSE NULL END)
    /COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN session_id ELSE NULL END) AS billing_click_rt
FROM t2;
```

| lander_click_rt | products_click_rt | mrfuzzy_click_rt | cart_click_rt | shipping_click_rt | billing_click_rt |
|---|---|---|---|---|---|
| 47.09% | 74.05% | 43.52% | 66.57% | 79.30% | 43.61% |

The CTRs in above query output shows the percentage of people clicks through at each step in the funnel flow from lander to billing. Among all steps in the funnel, lander, Mr. Fuzzy page, and the billing page have the lowest CTRs, which may need to be focused on to improve performance.

## Assignment 3.2: Analyzing Conversion Funnel Tests

```sql
-- Step 1: Find the starting point to frame the analysis for fair comparison
SELECT MIN(website_pageview_id) AS first_billing2_pv_id
FROM website_pageviews
WHERE pageview_url = '/billing-2'
-- first_billing2_pv_id: 53350

-- Step 2: Summary Sessions, Orders, and CVRs for each billing page
WITH t1 AS
(
    SELECT
        w.website_session_id AS session_id,
        w.pageview_url AS billing_version_seen,
        o.order_id
    FROM website_pageviews w LEFT JOIN orders o
        ON w.website_session_id = o.website_session_id
    WHERE w.website_pageview_id >= 53550 -- where test was live
        AND w.created_at < '2012-11-10' -- date of assignment
        AND w.pageview_url IN ('/billing','/billing-2')
)

SELECT
    billing_version_seen,
    COUNT(DISTINCT session_id) AS sessions,
    COUNT(DISTINCT order_id) AS Orders,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT session_id) AS billing_to_order_rt
FROM t1
GROUP BY 1;
```

| | billing_version_seen | sessions | Orders | billing_to_order_rt |
|---|---|---|---|---|
| ▶ | /billing | 657 | 300 | 0.4566 |
| | /billing-2 | 654 | 410 | 0.6269 |

Based on the query output, /billing-2 is converting customers much better than the previous billing page in that it has helped generate more orders and the conversion rate is substantially higher. The new website design might have improved the customer experience so that they are more willing to fill in billing formation and proceed to place an order.