

Maven Fuzzy Factory Database Overview

Maven Fuzzy Factory is an eCommerce retailer that specializes in SITUATION products. As a new eCommerce Database Analyst for the company, your role involves working closely with the CEO, the Head of Marketing, and the Website Manager to steer the business by analyzing and optimizing marketing channels, website performance, and product portfolio using SQL.

Introduction

In your role as an eCommerce Database Analyst at Maven Fuzzy Factory, you are responsible for extracting and analyzing data from the company's database to understand and communicate the story of its growth over the first eight months of operation. The CEO is due to present company performance metrics to the board, and this is an opportunity to showcase the company's analytical capabilities. This report will help you effectively communicate the growth story to the board and highlight your role as the data expert for the company.

Midterm Project

Project Overview

Maven Fuzzy Factory has been operational for approximately eight months, and your CEO needs to present company performance metrics to the board in the upcoming week. As the eCommerce Database Analyst, your primary responsibility is to extract and analyze data from the company's database to quantify the company's growth during this period and effectively communicate this growth story to the board.

Data Analysis Goals

1. **Quantify Growth:** Use SQL to extract and analyze website traffic and performance data from the Maven Fuzzy Factory database to measure the company's growth over the first eight months of operation.
2. **Create a Compelling Story:** Present the data in a way that tells the story of how the company achieved this growth, demonstrating your analytical capabilities.

Email from Cindy (November 27, 2012)

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

Cindy, the CEO, has reached out to you on November 27, 2012, requesting your assistance in preparing a presentation for the board meeting. The board's primary interest is gaining insights into the company's growth story during the initial eight months of operation. This presentation is also an opportunity to showcase the analytical capabilities of the company.

Now, you are tasked with leveraging your expertise in SQL to extract, analyze, and effectively communicate the company's growth story to the board. The next sections of your report will delve into the details of the data analysis and presentation.

Data Information

The Maven Fuzzy Factory database consists of six tables, each serving a specific purpose. These tables are as follows:

1. **website_sessions**

- Fields:
 - website_session_id (BIGINT UNSIGNED)
 - created_at (DATETIME)
 - user_id (BIGINT UNSIGNED)
 - is_repeat_session (SMALLINT UNSIGNED)
 - utm_source (VARCHAR)
 - utm_campaign (VARCHAR)
 - utm_content (VARCHAR)
 - device_type (VARCHAR)
 - http_referer (VARCHAR)
- Primary Key: website_session_id

2. **website_pageviews**

- Fields:
 - website_pageview_id (BIGINT UNSIGNED)
 - created_at (DATETIME)
 - website_session_id (BIGINT UNSIGNED)
 - pageview_url (VARCHAR)
- Primary Key: website_pageview_id

3. **products**

- Fields:
 - product_id (BIGINT UNSIGNED)
 - created_at (DATETIME)
 - product_name (VARCHAR)
- Primary Key: product_id

4. **orders**

- Fields:
 - order_id (BIGINT UNSIGNED)
 - created_at (DATETIME)
 - website_session_id (BIGINT UNSIGNED)
 - user_id (BIGINT UNSIGNED)
 - primary_product_id (SMALLINT UNSIGNED)
 - items_purchased (SMALLINT UNSIGNED)
 - price_usd (DECIMAL)
 - cogs_usd (DECIMAL)
- Primary Key: order_id

5. **order_items**

- Fields:

- order_item_id (BIGINT UNSIGNED)
- created_at (DATETIME)
- order_id (BIGINT UNSIGNED)
- product_id (SMALLINT UNSIGNED)
- is_primary_item (SMALLINT UNSIGNED)
- price_usd (DECIMAL)
- cogs_usd (DECIMAL)
- Primary Key: order_item_id

6. **order_item_refunds**

- Fields:
 - order_item_refund_id (BIGINT UNSIGNED)
 - created_at (DATETIME)
 - order_item_id (BIGINT UNSIGNED)
 - order_id (BIGINT UNSIGNED)
 - refund_amount_usd (DECIMAL)
- Primary Key: order_item_refund_id

These tables hold crucial data that will enable you to quantify the company's growth, optimize marketing channels, and measure website performance. In the upcoming sections, you will explore how to extract and analyze this data to create a compelling growth story for Maven Fuzzy Factory's presentation to the board.

Requests:

Request 1

Cindy has requested information to showcase the growth of Gsearch sessions and orders on a monthly basis. We need to extract monthly trends for Gsearch sessions and orders to demonstrate their impact on the company's growth.

SQL Query

```
SELECT Year(a.created_at) AS year,
       Month(a.created_at) AS month,
       Count(DISTINCT a.website_session_id) AS sessions,
       Count(DISTINCT b.order_id) AS orders,
       Round(Count(DISTINCT b.order_id) / Count(DISTINCT
a.website_session_id) * 100, 2) AS conv_rate
FROM website_sessions a
LEFT JOIN orders b
      ON b.website_session_id = a.website_session_id
WHERE a.created_at < '2012-11-27'
      AND a.utm_source = 'gsearch'
GROUP BY 1,
        2;
```

Query Explanation

In this SQL query:

- We select the year and month of the created_at column to group the data on a monthly basis.
- We count the distinct website sessions associated with Gsearch as "sessions."
- We count the distinct orders linked to these sessions as "orders."
- We calculate the conversion rate as the percentage of orders divided by sessions, rounded to two decimal places.

We use a LEFT JOIN to ensure that all Gsearch sessions are included, even if there were no orders associated with them.

We filter the data to include only records created before November 27, 2012, and where the utm_source is 'gsearch.' The results are grouped by year and month.

Answer

The query returns the following result:

Year	Month	sessions	orders	cvr
2012	3	1860	60	3.23
2012	4	3574	92	2.57
2012	5	3410	97	2.84
2012	6	3578	121	3.38
2012	7	3811	145	3.80
2012	8	4877	184	3.77
2012	9	4491	188	4.19
2012	10	5534	234	4.23
2012	11	8889	373	4.20

Answer Interpretation

The table provides a monthly breakdown of Gsearch-related website sessions and orders. It is clear that both sessions and orders have been consistently increasing over the months. The conversion rate, which represents the percentage of orders relative to sessions, shows positive growth as well.

This data demonstrates the significant impact of Gsearch on the company's growth story, making it a valuable marketing channel. This information can be used to showcase the company's growth to the board as requested.

Request 2

Cindy has requested a monthly trend analysis of Gsearch, specifically splitting out nonbrand and brand campaigns separately. The goal is to determine whether the brand campaign is gaining traction, as this could be a positive story to present.

SQL Query

```
SELECT Year(a.created_at) AS year,
       Month(a.created_at) AS month,
       Count(DISTINCT CASE
                WHEN a.utm_campaign = 'nonbrand' THEN
                a.website_session_id
                ELSE NULL
            end) AS non_brand_sessions,
       Count(DISTINCT CASE
                WHEN a.utm_campaign = 'nonbrand' THEN b.order_id
                ELSE NULL
            end) AS non_brand_orders,
       Count(DISTINCT CASE
                WHEN a.utm_campaign = 'brand' THEN a.website_session_id
                ELSE NULL
            end) AS brand_sessions,
       Count(DISTINCT CASE
                WHEN a.utm_campaign = 'brand' THEN b.order_id
                ELSE NULL
            end) AS brand_orders
FROM   website_sessions a
       LEFT JOIN orders b
            ON b.website_session_id = a.website_session_id
WHERE  a.created_at < '2012-11-27'
       AND a.utm_source = 'gsearch'
GROUP BY 1,
         2;
```

Query Explanation

In this SQL query:

- We select the year and month of the created_at column to group the data on a monthly basis.
- We use conditional statements to count nonbrand sessions, nonbrand orders, brand sessions, and brand orders separately. We determine this based on the utm_campaign value.
- We perform a LEFT JOIN between the website_sessions and orders tables using the website_session_id to ensure all Gsearch sessions are included, whether they have orders or not.

The data is filtered to include records created before November 27, 2012, and where the utm_source is 'gsearch'. The results are grouped by year and month.

Answer

The query returns the following result:

Year	month	nonbrand_sessions	nonbrand_orders	brand_sessions	brand_orders
2012	3	1852	60	8	0
2012	4	3509	86	65	6
2012	5	3295	91	115	6
2012	6	3439	114	139	7
2012	7	3660	136	151	9
2012	8	4673	174	204	10
2012	9	4227	172	264	16
2012	10	5197	219	337	15
2012	11	8506	356	383	17

Answer Interpretation

The table provides a monthly breakdown of Gsearch sessions and orders, categorizing them into nonbrand and brand campaigns. This data analysis shows that both nonbrand and brand campaigns have seen growth over the eight-month period.

For brand campaigns, although the numbers are smaller compared to nonbrand campaigns, they have also been steadily increasing, indicating positive traction. This trend could be used as a compelling story to showcase the success and growth of the brand campaign during the presentation to the board.

Request 3

Cindy has requested a deep dive into the nonbrand campaign for Gsearch. Specifically, you need to provide a monthly breakdown of sessions and orders, splitting the data by device type. The goal is to demonstrate a comprehensive understanding of traffic sources to impress the board with the company's analytical capabilities.

SQL Query

```
SELECT Year(a.created_at) AS year,
       Month(a.created_at) AS month,
       Count(DISTINCT a.website_session_id) AS sessions,
       Count(DISTINCT b.order_id) AS orders,
       Count(DISTINCT CASE
         WHEN a.device_type = 'mobile' THEN a.website_session_id
         ELSE NULL
       end) AS mobile_sessions,
       Count(DISTINCT CASE
         WHEN a.device_type = 'desktop' THEN a.website_session_id
         ELSE NULL
       end) AS desktop_sessions,
       Count(DISTINCT CASE
         WHEN a.device_type = 'mobile' THEN b.order_id
         ELSE NULL
       end) AS mobile_orders,
       Count(DISTINCT CASE
         WHEN a.device_type = 'desktop' THEN b.order_id
         ELSE NULL
       end) AS desktop_orders
FROM   website_sessions a
       LEFT JOIN orders b
         ON b.website_session_id = a.website_session_id
WHERE  a.created_at < '2012-11-27'
       AND a.utm_source = 'gsearch'
       AND a.utm_campaign = 'nonbrand'
GROUP BY 1,
         2;
```

Query Explanation

In this SQL query:

- We select the year and month of the created_at column to group the data on a monthly basis.
- We count the distinct website sessions and orders.
- We use conditional statements to count sessions and orders separately for mobile and desktop devices based on the device_type value.
- We perform a LEFT JOIN between the website_sessions and orders tables using the website_session_id to ensure all Gsearch sessions are included, whether they have orders or not.

The data is filtered to include records created before November 27, 2012, where the utm_source is 'gsearch,' and the utm_campaign is 'nonbrand.' The results are grouped by year and month.

Answer

The query returns the following result:

Year	month	sessions	orders	mobile_sessions	desktop_sessions	mobile_orders	desktop_orders
2012	3	1852	60	724	1128	10	50
2012	4	3509	86	1370	2139	11	75
2012	5	3295	91	1019	2276	8	83
2012	6	3439	114	766	2673	8	106
2012	7	3660	136	886	2774	14	122
2012	8	4673	174	1158	3515	9	165
2012	9	4227	172	1056	3171	17	155
2012	10	5197	219	1263	3934	18	201
2012	11	8506	356	2049	6457	33	323

Answer Interpretation

The table presents a monthly breakdown of Gsearch sessions and orders for the nonbrand campaign, categorized by mobile and desktop devices. This analysis highlights the company's in-depth understanding of its traffic sources, showcasing the traffic patterns for nonbrand Gsearch sessions on different devices.

The data demonstrates the growth and composition of nonbrand Gsearch sessions and orders, allowing the board to appreciate the company's analytical capabilities and its ability to track and optimize traffic sources effectively. This information can be used to impress the board during the presentation.

Request 4

Cindy is concerned about a board member's worries regarding the high percentage of traffic coming from Gsearch. She wants to compare monthly trends for Gsearch alongside monthly trends for each of the company's other marketing channels. The objective is to provide a comprehensive view of traffic sources and reassure the board about the diversity of traffic channels.

SQL Query

In the first part of the query, we identify the various UTM sources and HTTP referrers to understand the traffic sources: In the second part of the query, we analyze the monthly trends for different traffic sources, including Gsearch and other channels:

```
-- first, finding the various utm sources and referers to see the traffic we're getting
SELECT DISTINCT utm_source,
                 utm_campaign,
                 http_referrer
FROM   website_sessions
WHERE  created_at < '2012-11-27';

SELECT Year(a.created_at) AS year,
       Month(a.created_at) AS month,
       Count(DISTINCT CASE
                 WHEN a.utm_source = 'gsearch' THEN a.website_session_id
                 ELSE NULL
               end) AS gsearch_paid_sessions,
       Count(DISTINCT CASE
                 WHEN a.utm_source = 'bsearch' THEN a.website_session_id
                 ELSE NULL
               end) AS bsearch_paid_sessions,
       Count(DISTINCT CASE
                 WHEN a.utm_source IS NULL
                      AND a.http_referrer IS NOT NULL THEN
                   a.website_session_id
                 ELSE NULL
               end) AS organic_search_sessions,
       Count(DISTINCT CASE
                 WHEN a.utm_source IS NULL
                      AND a.http_referrer IS NULL THEN
                   a.website_session_id
                 ELSE NULL
               end) AS direct_type_in_sessions
FROM   website_sessions a
       LEFT JOIN orders b
           ON b.website_session_id = a.website_session_id
WHERE  a.created_at < '2012-11-27'
GROUP BY 1,
         2;
```

Query Explanation

In this SQL query:

- We select the year and month of the created_at column to group the data on a monthly basis.
- We use conditional statements to count sessions for Gsearch, Bsearch, organic search, and direct type-ins.
- We perform a LEFT JOIN between the website_sessions and orders tables using the website_session_id to ensure all sessions are included, whether they have orders or not.

The data is filtered to include records created before November 27, 2012. The results are grouped by year and month.

Answer (Part 1)

The result shows distinct UTM sources, UTM campaigns, and HTTP referrers:

utm_source	utm_campaign	http_referrer
gsearch	nonbrand	https://www.gsearch.com
na	na	na
gsearch	brand	https://www.gsearch.com
na	na	https://www.gsearch.com
bsearch	brand	https://www.bsearch.com
na	na	https://www.bsearch.com
bsearch	nonbrand	https://www.bsearch.com

Answer (Part 2)

The query returns the following result:

year	month	gsearch_paid_sessions	bsearch_paid_sessions	organic_search_sessions	direct_type_in_sessions
2012	3	1860	2	8	9
2012	4	3574	11	78	71
2012	5	3410	25	150	151
2012	6	3578	25	190	170
2012	7	3811	44	207	187
2012	8	4877	705	265	250
2012	9	4491	1439	331	285
2012	10	5534	1781	428	440
2012	11	8889	2840	536	485

Answer Interpretation

The table provides a monthly breakdown of sessions for different traffic sources, including Gsearch, Bsearch, organic search, and direct type-ins. By presenting data from multiple channels, it showcases the diversity of traffic sources, which can help alleviate concerns about a high percentage of traffic from Gsearch.

This analysis allows the company to demonstrate a comprehensive understanding of its traffic sources, reinforcing the board's confidence in the company's ability to manage and diversify its marketing channels effectively. This information can be used to address concerns during the board meeting.

Request 5

Cindy wants to showcase the story of website performance improvements over the first 8 months by presenting session to order conversion rates on a monthly basis.

SQL Query

```
SELECT Year(a.created_at) AS year,
       Month(a.created_at) AS month,
       Count(DISTINCT a.website_session_id) AS sessions,
       Count(DISTINCT b.order_id) AS orders,
       Round(Count(DISTINCT b.order_id) / Count(DISTINCT
a.website_session_id) * 100, 2) AS conv_rate
FROM website_sessions a
LEFT JOIN orders b
      ON b.website_session_id = a.website_session_id
WHERE a.created_at < '2012-11-27'
GROUP BY 1,
        2;
```

Query Explanation

In this SQL query:

- We select the year and month of the created_at column to group the data on a monthly basis.
- We count the distinct website sessions and orders.
- We calculate the session to order conversion rate by dividing the count of orders by the count of sessions and multiplying by 100 to express it as a percentage.

The data is filtered to include records created before November 27, 2012. The results are grouped by year and month.

Answer

The query returns the following result:

year	month	sessions	orders	conv_rate
2012	3	1879	60	3.19
2012	4	3734	99	2.65
2012	5	3736	108	2.89
2012	6	3963	140	3.53
2012	7	4249	169	3.98
2012	8	6097	228	3.74
2012	9	6546	287	4.38
2012	10	8183	371	4.53
2012	11	12750	561	4.40

Answer Interpretation

The table provides a monthly overview of website performance improvements over the course of the first 8 months. It includes the number of sessions, the number of orders, and the session to order conversion rate expressed as a percentage.

The data shows an increase in both sessions and orders over the period, with the conversion rate gradually improving. This information allows the company to tell a positive story about its website's performance and growth. The board can gain confidence in the company's ability to optimize its website and convert more sessions into orders.

Request 6

Cindy wants to estimate the revenue earned from the "gsearch lander test" by looking at the increase in conversion rate (CVR) during the test period (June 19 – July 28). The calculation should consider nonbrand sessions and revenue generated since the end of the test.

SQL Query

```
-- first entry id of lander-1
SELECT Min(website_pageview_id) AS first_test_pv
FROM website_pageviews
WHERE pageview_url = '/lander-1';

-- 23504 is the answer
WITH cte1
  AS (SELECT a.website_session_id,
            Min(a.website_pageview_id) AS min_pageview_id
      FROM website_pageviews a
      INNER JOIN website_sessions b
            ON b.website_session_id = a.website_session_id
            AND b.created_at < '2012-07-28'
            -- prescribed by the assignment
            AND a.website_pageview_id >= 23504 -- first page_view
            AND utm_source = 'gsearch'
            AND utm_campaign = 'nonbrand'
      GROUP BY a.website_session_id),
  cte2
  AS (SELECT a.website_session_id,
            b.pageview_url AS landing_page
      FROM cte1 a
      LEFT JOIN website_pageviews b
            ON b.website_pageview_id = a.min_pageview_id
      WHERE b.pageview_url IN ( '/home', '/lander-1' )),
  cte3
  AS (SELECT a.website_session_id,
            a.landing_page,
            b.order_id AS order_id
      FROM cte2 a
      LEFT JOIN orders b
            ON b.website_session_id = a.website_session_id)
SELECT landing_page,
       Count(DISTINCT website_session_id) AS sessions,
       Count(DISTINCT order_id)           AS orders,
       Round(( Count(DISTINCT order_id) / Count(DISTINCT website_session_id) *
              100 ), 2
             )
       AS conv_rate
FROM cte3
GROUP BY 1;

-- finding the most recent pageview for gsearch nonbrand where the traffic was sent to /home
SELECT Max(a.website_session_id) AS most_recent_gsearch_nonbrand_home_pageview
FROM website_sessions a
LEFT JOIN website_pageviews b
      ON b.website_session_id = a.website_session_id
WHERE utm_source = 'gsearch'
      AND utm_campaign = 'nonbrand'
      AND pageview_url = '/home'
      AND a.created_at < '2012-11-27';

-- 17145
SELECT Count(website_session_id) AS sessions_since_test
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';
-- 22,972 website sessions since the test
-- X .0087 incremental conversion = 202 incremental orders since 7/29
-- roughly 4 months, so roughly 50 extra orders per month. Not bad!
```

Query Explanation

In this SQL query:

- We first find the entry ID for the "gsearch lander test" page ('/lander-1').

- We use Common Table Expressions (CTEs) to gather data for the test, including sessions, landing pages, and orders.
- The data is filtered to focus on the specified date range and UTM conditions.

The result provides sessions, orders, and conversion rates for the home and lander-1 pages.

Answer (Part 1)

The query returns the following result:

Landing Page	Sessions	Orders	Conversion Rate
/home	2261	72	3.18
/lander-1	2316	94	4.06

In the second part of the query, we estimate the number of website sessions since the end of the test that belong to the "gsearch nonbrand" category with the "home" page as the landing page:

In this SQL query:

- We find the most recent pageview for "gsearch nonbrand" where the traffic was sent to the "/home" page.
- We calculate the number of website sessions since the end of the test that meet the specified conditions.

Answer (Part 2)

The query returns the following result:

Sessions Since Test
22,972

Answer Interpretation

To estimate the incremental value from the "gsearch lander test," we first compare the conversion rates for the "home" and "lander-1" pages. The "lander-1" page had a higher conversion rate, indicating the effectiveness of the test.

To estimate revenue, we consider the number of sessions since the test that meet the "gsearch nonbrand" and "/home" landing page criteria, resulting in 22,972 sessions. We estimate incremental orders by multiplying this count by the incremental conversion rate. The estimated incremental orders are roughly 202 since July 29, which equates to approximately 50 extra orders per month over a four-month period.

This analysis helps showcase the positive impact of the "gsearch lander test" in terms of conversion rate improvement and potential revenue increase. It can be used to demonstrate the value of such tests to the board.

Request 7

Cindy wants to create a full conversion funnel for each of the two pages, /home and /lander-1, showing the progression from these pages to orders during the specified time period (June 19 – July 28).

SQL Query

In this query, we create a temporary table called "funnel" and build the conversion funnel by identifying which pages users visited during their sessions and tracking the flow from the landing pages to orders:

```
create TEMPORARY table funnel
with
cte1 as (SELECT
  a.website_session_id,
  b.pageview_url,
  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 a
  LEFT JOIN website_pageviews b
    ON a.website_session_id = b.website_session_id
WHERE a.utm_source = 'gsearch'
  AND a.utm_campaign = 'nonbrand'
  AND a.created_at < '2012-07-28'
  AND a.created_at > '2012-06-19'
ORDER BY
  a.website_session_id,
  b.created_at)
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 cte1
GROUP BY 1;
```

```
-- overall numbers
SELECT
  CASE
    WHEN saw_homepage = 1 THEN 'saw_homepage'
    WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
    ELSE 'ERROR 404!'
  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,
  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 funnel
GROUP BY 1;
```



```
-- clickthrough rate
SELECT
  CASE
    WHEN saw_homepage = 1 THEN 'saw_homepage'
    WHEN saw_custom_lander = 1 THEN 'saw_custom_lander'
    ELSE 'ERROR 404!'
  END AS segment,
  COUNT(DISTINCT CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL
END)/COUNT(DISTINCT website_session_id) AS lander_click_rt,

  COUNT(DISTINCT CASE WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT
CASE WHEN product_made_it = 1 THEN website_session_id ELSE NULL END) AS products_click_rt,

  COUNT(DISTINCT CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT CASE
WHEN mrfuzzy_made_it = 1 THEN website_session_id ELSE NULL END) AS mrfuzzy_click_rt,

  COUNT(DISTINCT CASE WHEN shipping_made_it = 1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT
CASE WHEN cart_made_it = 1 THEN website_session_id ELSE NULL END) AS cart_click_rt,

  COUNT(DISTINCT CASE WHEN billing_made_it = 1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT
CASE WHEN shipping_made_it = 1 THEN website_session_id ELSE NULL END) AS shipping_click_rt,

  COUNT(DISTINCT CASE WHEN thankyou_made_it = 1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT
CASE WHEN billing_made_it = 1 THEN website_session_id ELSE NULL END) AS billing_click_rt

FROM funnel
GROUP BY 1;
```

Query Explanation

In this SQL query:

- We create a temporary table "funnel" using Common Table Expressions (CTEs) to track users' progression through the pages, including homepage, custom lander, products page, Mr. Fuzzy page, cart page, shipping page, billing page, and thank-you page.
- We calculate overall numbers, including sessions, and the number of users who progress to different steps in the funnel.
- We also calculate clickthrough rates for each step of the funnel.

Answer

The query returns the following results:

Overall Numbers:

Segment	Sessions	to Products	to Mr. Fuzzy	to Cart	to Shipping	to Billing	to Thank You
Saw Custom Lander	2,316	1,083	772	348	231	197	94
Saw Homepage	2,261	942	684	296	200	168	72

Clickthrough Rates:

segment	lander_click_rt	products_click_rt	mrfuzzy_click_rt	cart_click_rt	shipping_click_rt	billing_click_rt
Saw Custom Lander	0.4676	0.7128	0.4508	0.6638	0.8528	0.4772
Saw Homepage	0.4166	0.7261	0.4327	0.6757	0.8400	0.4286

Answer Interpretation

The table shows the progression of sessions from the landing pages (/home and /lander-1) through various steps in the conversion funnel. It also provides clickthrough rates for

each step. The results help analyze user behavior and identify where users drop off or continue through the funnel.

For instance, it's evident that the "Saw Custom Lander" segment has a higher clickthrough rate for the "Products" and "Mr. Fuzzy" steps compared to the "Saw Homepage" segment. This information can be valuable for optimizing the website's user experience and increasing conversions.

Request 8

Cindy wants to quantify the impact of the billing test that took place from September 10 to November 10. Specifically, she's interested in understanding the revenue per billing page session during the test period and then identifying the number of billing page sessions for the past month to determine the monthly impact.

SQL Query

In this query, we calculate the revenue per billing page session during the test period for two billing page versions and then count the number of billing page sessions in the past month:

```
WITH cte1
  AS (SELECT a.website_session_id,
            a.pageview_url AS billing_version_seen,
            b.order_id,
            b.price_usd
      FROM website_pageviews a
      LEFT JOIN orders b
            ON b.website_session_id = a.website_session_id
      WHERE a.created_at > '2012-09-10' -- prescribed in assignment
            AND a.created_at < '2012-11-10' -- prescribed in assignment
            AND a.pageview_url IN ( '/billing', '/billing-2' ))
SELECT billing_version_seen,
       Count(DISTINCT website_session_id) AS sessions
       ,
       Round(Sum(price_usd) / Count(DISTINCT website_session_id), 2) AS
       revenue_per_billing_page_seen
FROM   cte1
GROUP BY 1;

-- $22.83 revenue per billing page seen for the old version
-- $31.34 for the new version
-- LIFT: $8.51 per billing page view

SELECT Count(website_session_id) AS billing_sessions_past_month
FROM   website_pageviews
WHERE  pageview_url IN ( '/billing', '/billing-2' )
       AND created_at BETWEEN '2012-10-27' AND '2012-11-27'; -- past month

-- 1,194 billing sessions past month
-- LIFT: $8.51 per billing session
-- VALUE OF BILLING TEST: $10,160 over the past month
```

Query Explanation

In this SQL query:

- We calculate revenue per billing page session during the test period for two billing page versions, /billing and /billing-2. We first join the website_pageviews and orders tables to calculate this.
- We also count the number of billing page sessions in the past month (from October 27, 2012, to November 27, 2012).

Answer

The query returns the following results:

Revenue per Billing Page Session During the Test:

Billing Version	Sessions	Revenue per Billing Page Session
/billing	657	\$22.83
/billing-2	654	\$31.34

The lift generated by the test (difference in revenue per billing page session):

- For the old version (/billing): \$22.83
- For the new version (/billing-2): \$31.34
- Lift: \$8.51 per billing page view

Number of Billing Page Sessions in the Past Month: 1,193 sessions

Answer Interpretation

The analysis shows the impact of the billing test during the test period. The new billing page version (/billing-2) generated higher revenue per billing page session compared to the old version (/billing).

In the past month, there were 1,193 billing page sessions. The calculated lift of \$8.51 per billing page session indicates the additional value generated by the new billing page version during the test period. The total value of the billing test is estimated at \$10,160 over the past month. This information helps quantify the test's impact on revenue.

Conclusion

In this analysis, we've delved into various aspects of website performance and marketing campaigns, aiming to provide valuable insights to improve decision-making and demonstrate the effectiveness of different strategies. Here's a summary of our findings:

Gsearch Performance

Request 1 - We analyzed monthly trends for Gsearch sessions and orders. The data showed consistent growth in sessions and orders from March to November 2012. The conversion rate remained steady at around 4%, indicating a strong performance.

Request 2 - We further segmented Gsearch performance into non-brand and brand campaigns. The analysis revealed that both non-brand and brand campaigns were successful, with a clear upward trend in sessions and orders.

Request 3 - Focusing on non-brand Gsearch, we explored performance by device type. Mobile sessions and orders showed consistent growth, emphasizing the importance of optimizing for mobile traffic.

Traffic Channels

Request 4 - We examined the performance of various traffic channels, including Gsearch, Bsearch, organic search, and direct traffic. The data showed Gsearch as the primary driver of sessions, with increasing sessions from July to November.

Website Performance

Request 5 - We analyzed the session-to-order conversion rates for the first eight months. While there were fluctuations, the overall conversion rate improved, demonstrating the effectiveness of website performance improvements.

A/B Testing

Request 6 - We estimated the incremental revenue generated by the Gsearch lander test, revealing that the test led to roughly 50 extra orders per month, contributing positively to revenue.

Request 7 - For the landing page test, we built a conversion funnel to track user progress from the homepage to the thank-you page. The analysis showed that a significant number of users progressed through the funnel, indicating the effectiveness of the test.

Request 8 - We quantified the impact of the billing test by comparing the revenue per billing page session for different billing page versions. The new billing page version outperformed the old one, leading to additional revenue per session and an estimated \$10,160 impact over the past month.

Key Takeaways

1. Gsearch was a major driver of website traffic and orders. Both non-brand and brand campaigns were successful in increasing sessions and orders.
2. Mobile traffic from Gsearch was significant, emphasizing the need for mobile optimization.
3. A/B testing, such as the lander and billing page tests, proved valuable in generating incremental revenue and improving user experience.
4. The website's overall performance improved over the months, leading to higher session-to-order conversion rates.

5. Conversion funnels are effective tools for tracking user progress through the website, providing insights into user behavior.

These insights should guide decision-making and strategy development to further enhance website performance and marketing effectiveness. Regular monitoring and analysis of these key metrics will be crucial to maintaining and improving overall performance.