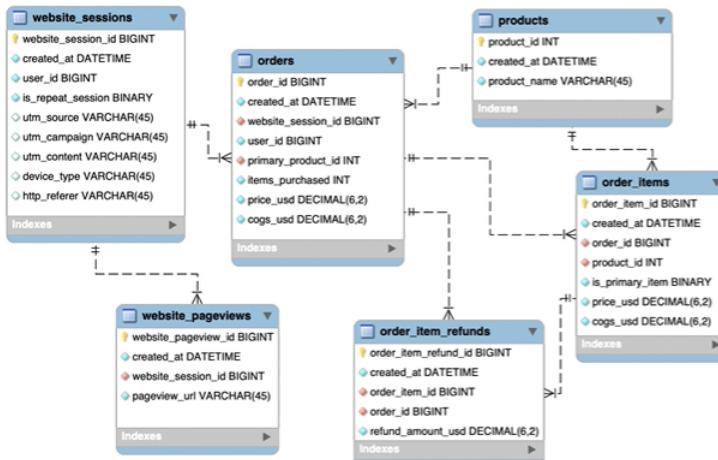




Advance MySQL Udemy Course

Maven MySQL Advanced Course in Udemy



We will be working with six related tables, which contain eCommerce data about:

- Website Activity
- Products
- Orders and Refunds

We'll use MySQL to understand how customers access and interact with the site, analyze landing page performance and conversion, and explore product-level sales.

1. Where all the website sessions are coming from?

NEW MESSAGE
April 12, 2012

From: Cindy Sharp (CEO)
Subject: Site traffic breakdown

Good morning,

We've been live for almost a month now and we're starting to generate sales. Can you help me understand where the bulk of our website sessions are coming from, through yesterday?

I'd like to see a breakdown by **UTM source**, **campaign** and **referring domain** if possible. Thanks!

-Cindy

Result Preview

utm_source	utm_campaign	http_referer	sessions
3613			
28			
27			
26			
7			
7			

```

SELECT
    utm_source,
    utm_campaign,
    http_referer,
    COUNT(DISTINCT website_session_id) as Sessions
  
```

```

FROM website_sessions
WHERE created_at < '2012-04-12'
GROUP BY
    utm_source,
    utm_campaign,
    http_referer
ORDER BY 4 DESC

```

Output

	utm_source	utm_campaign	http_referer	Sessions
▶	gsearch	nonbrand	https://www.gsearch.com	3613
	NULL	NULL	NULL	28
	NULL	NULL	https://www.gsearch.com	27
	gsearch	brand	https://www.gsearch.com	26
	NULL	NULL	https://www.bsearch.com	7
	bsearch	brand	https://www.bsearch.com	7

2. Analyze deeper on the gsearch and nonbrand, which drives the most sessions.

The email content is as follows:

Hi there,

Sounds like gsearch nonbrand is our major traffic source, but we need to understand if those sessions are driving sales.

Could you please calculate the conversion rate (CVR) from session to order? Based on what we're paying for clicks, we'll need a CVR of at least 4% to make the numbers work.

If we're much lower, we'll need to reduce bids. If we're higher, we can increase bids to drive more volume.

Thanks, Tom

The result grid screenshot shows the following data:

sessions	orders	session_to_order_conv_rate
3613	112	2.8755

```

SELECT
    COUNT(distinct website_sessions.website_session_id) as Sessions,
    COUNT(DISTINCT orders.order_id) as Orders,
    (COUNT(DISTINCT orders.order_id)/COUNT(distinct website_sessions.website_session_id))*100 as Coverstion_rate
FROM website_sessions
LEFT JOIN orders
ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-04-14'
AND website_sessions.utm_campaign='nonbrand'
AND website_sessions.utm_source='gsearch';

```

Output

	Sessions	Orders	Coverstion_rate
	3895	112	2.8755

3. Pivoting product ID count of single purchase and two purchase and total purchase

CASE & COUNT

- The `orders_w_1_item` column is created by counting `order_id` values for records which have a value of 1 in the `items_purchased` column
- This method of counting records where a condition is true can be incredibly useful!

Showing rows 0 - 24 (of 24 total)

PRO TIP:
Use `GROUP BY` to define your row labels, and `CASE` to pivot to columns

MySQL QUERY IN ACTION:

```

1 * use mavenfuzzyfactory;
2 * 
3 * SELECT
4 *   primary_product_id,
5 *   COUNT(DISTINCT CASE WHEN items_purchased = 1 THEN order_id ELSE NULL END) AS orders_w_1_item,
6 *   COUNT(DISTINCT CASE WHEN items_purchased = 2 THEN order_id ELSE NULL END) AS orders_w_2_items,
7 *   COUNT(DISTINCT order_id) AS total_orders
8 * FROM orders
9 * WHERE order_id BETWEEN 31000 AND 32000
10 GROUP BY 1

```

You can identify which column to GROUP BY quickly by specifying it's column order within the SELECT statement. In this case, `primary_product_id` is column 1, so we can simply write "GROUP BY 1".

```

SELECT
    primary_product_id,
    count(CASE WHEN items_purchased = 1 THEN order_id ELSE NULL END) AS count_single_purchase,
    count(CASE WHEN items_purchased = 2 THEN order_id ELSE NULL END) AS count_two_purchase,
    count(distinct order_id) as total_order
FROM mavenfuzzyfactory.orders
WHERE order_id between 31000 AND 32000
GROUP BY 1

```

Output

	primary_product_id	count_single_purchase	count_two_purchase	total_order
▶	1	406	256	662
	2	99	38	137
	3	73	44	117
	4	75	10	85

After reducing the campaign cost of nonbrand gsearch, sessions trend

NEW MESSAGE
May 12, 2012

From: Tom Parmesan (Marketing Director)
Subject: Gsearch volume trends

Hi there,

Based on your conversion rate analysis, we **bid down gsearch nonbrand** on 2012-04-15.

Can you pull **gsearch nonbrand** trended session volume, by week, to see if the bid changes have caused volume to drop at all?

Thanks, Tom

Result Preview

Result Grid **Filter**

week_start_date	sessions
2012-03-19	
2012-03-25	
2012-04-01	
2012-04-08	
2012-04-15	
2012-04-22	
2012-04-29	
2012-05-06	

```

SELECT
    week(created_at) as Week_num,
    MIN(date(created_at)) as week_start,
    COUNT(DISTINCT website_session_id) as Sessions
FROM website_sessions
WHERE created_at < '2012-05-10'
AND utm_campaign = 'nonbrand'
AND utm_source = 'gsearch'
GROUP BY 1;

```

Output

	Week_num	week_start	Sessions
13	2012-03-25	956	
14	2012-04-01	1152	
15	2012-04-08	983	
16	2012-04-15	621	
17	2012-04-22	594	
18	2012-04-29	681	
19	2012-05-06	399	

Sessions and conversion rate by device type

Result Preview

device_type	sessions	orders	session_to_order_conv_rate
mobile	[redacted]	[redacted]	[redacted]
desktop	[redacted]	[redacted]	[redacted]

```

SELECT
    DISTINCT website_sessions.device_type,
    COUNT(DISTINCT website_sessions.website_session_id) as Sessions,
    COUNT(DISTINCT orders.order_id) as Orders,
    (COUNT(DISTINCT orders.order_id)/COUNT(DISTINCT website_sessions.website_session_id))*100 AS Conversion_rate
FROM website_sessions
LEFT JOIN orders
ON orders.website_session_id = website_sessions.website_session_id
WHERE website_sessions.created_at < '2012-05-11'
AND utm_campaign = 'nonbrand'
AND utm_source = 'gsearch'
GROUP BY 1;

```

Output

	device_type	Sessions	Orders	Conversion_rate
▶	desktop	3911	146	3.7331
	mobile	2492	24	0.9631

@August 26, 2022

Creating temporary table

Retrieve the landing pages session count

```
CREATE TEMPORARY TABLE first_pageview
SELECT
    website_session_id,
    MIN(website_pageview_id) as Landing_page
FROM website_pageviews
WHERE website_pageview_id < 1000
GROUP BY 1;

SELECT
    website_pageviews.pageview_url as Mainpage,
    COUNT(DISTINCT website_pageviews.website_session_id) as Sessions
FROM first_pageview
LEFT JOIN website_pageviews
    ON first_pageview.Landing_page = website_pageviews.website_pageview_id
GROUP BY 1;
```

@August 27, 2022

NEW MESSAGE
June 09, 2012

From: Morgan Rockwell (Website Manager)
Subject: Top Website Pages

Hi there!
I'm Morgan, the new Website Manager.
Could you help me get my head around the site by pulling the **most-viewed website pages, ranked by session volume?**
Thanks!
-Morgan

Result Preview

pageview_url	sessions
www.example.com/home	12345
www.example.com/about	8901
www.example.com/contact	5678
www.example.com/services	4321
www.example.com/pricing	3210
www.example.com/testimonials	2109
www.example.com/faqs	1234

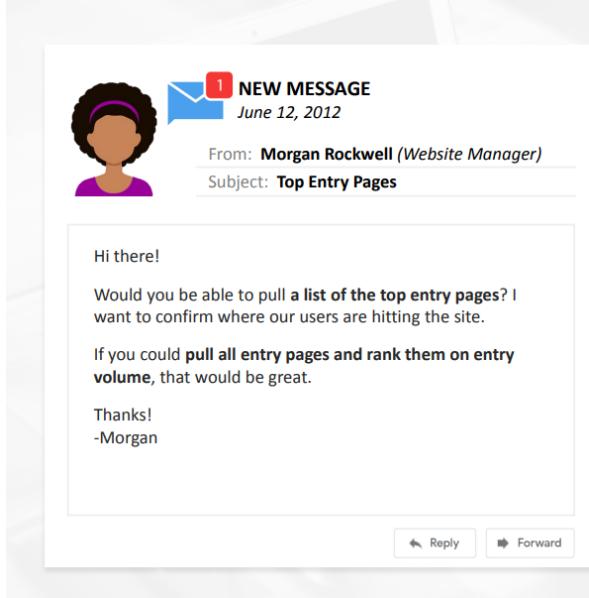
```

SELECT
    pageview_url,
    COUNT(DISTINCT website_pageview_id) as Sessions
FROM website_pageviews
WHERE created_at < '2012-06-09'
GROUP BY 1
ORDER BY 2 DESC;

```

Output

	pageview_url	Sessions
▶	/home	10403
	/products	4239
	/the-original-mr-fuzzy	3037
	/cart	1306
	/shipping	869
	/billing	716
	/thank-you-for-your-order	306



NEW MESSAGE
June 12, 2012

From: **Morgan Rockwell (Website Manager)**
Subject: **Top Entry Pages**

Hi there!
Would you be able to pull a list of the top entry pages? I want to confirm where our users are hitting the site.
If you could pull all entry pages and rank them on entry volume, that would be great.
Thanks!
-Morgan

Result Preview

landing_page	sessions_hitting_this_landing_page
▶ /home	
/products	
/the-original-mr-fuzzy	

```

CREATE TEMPORARY TABLE firstpageview1
SELECT
    website_session_id,
    MIN(website_pageview_id) as pag_view_id
FROM website_pageviews
WHERE created_at < '2012-06-12'
GROUP BY 1;

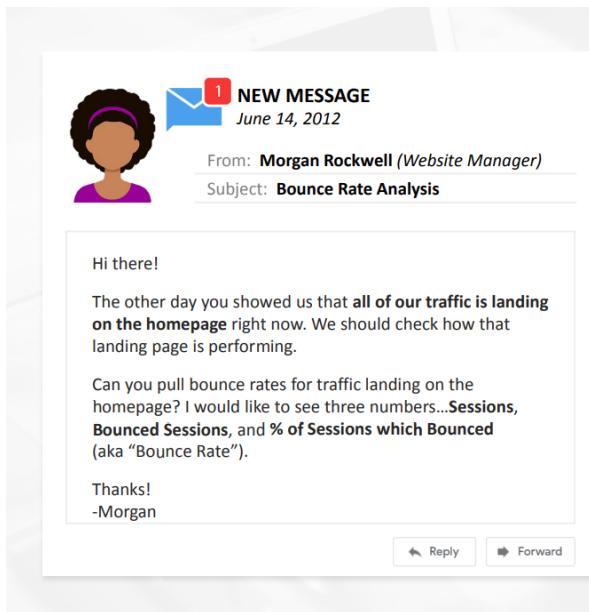
SELECT
    website_pageviews.pageview_url,
    COUNT(DISTINCT firstpageview1.pag_view_id) AS Sessions
FROM firstpageview1
LEFT JOIN website_pageviews

```

```
ON firstpageview1.pag_view_id = website_pageviews.website_pageview_id
GROUP BY website_pageviews.pageview_url;
```

Output

	pageview_url	Sessions
▶	/home	10714



NEW MESSAGE
June 14, 2012

From: **Morgan Rockwell (Website Manager)**
Subject: **Bounce Rate Analysis**

Hi there!
The other day you showed us that **all of our traffic is landing on the homepage** right now. We should check how that landing page is performing.
Can you pull bounce rates for traffic landing on the homepage? I would like to see three numbers...**Sessions, Bounced Sessions, and % of Sessions which Bounced** (aka "Bounce Rate").
Thanks!
-Morgan

Reply Forward

Result Preview

Result Grid	Filter Rows:
sessions bounced_sessions bounce_rate	11044 6536 0.5918

```
CREATE TEMPORARY TABLE landing_page1
SELECT
    website_session_id,
    MIN(website_pageview_id) as landing_page_id
FROM website_pageviews
WHERE created_at < '2012-06-14'
GROUP BY 1;

CREATE TEMPORARY TABLE first_sessions
SELECT
    website_pageviews.website_session_id,
    website_pageviews.pageview_url
FROM landing_page1
LEFT JOIN website_pageviews
    ON landing_page1.landing_page_id = website_pageviews.website_pageview_id;

SELECT * FROM first_sessions;

CREATE TEMPORARY TABLE bounce_sessions
SELECT
    first_sessions.website_session_id,
    first_sessions.pageview_url,
    COUNT(DISTINCT website_pageviews.website_pageview_id) AS Sessions
FROM first_sessions
LEFT JOIN website_pageviews
    ON website_pageviews.website_session_id = first_sessions.website_session_id
GROUP BY 1, 2;
```

```

HAVING COUNT(website_pageviews.website_pageview_id) = 1;

SELECT
    COUNT(DISTINCT first_sessions.website_session_id) AS Sessions,
    COUNT(DISTINCT bounce_sessions.website_session_id) AS bounced_sessions,
    COUNT(DISTINCT bounce_sessions.website_session_id)/COUNT(DISTINCT first_sessions.website_session_id) AS Bounce_rate
FROM first_sessions
LEFT JOIN bounce_sessions
    ON first_sessions.website_session_id = bounce_sessions.website_session_id
-- ORDER BY first_sessions.website_session_id

```

Output

	Sessions	bounced_sessions	Bounce_rate
▶	11048	6538	0.5918

@September 1, 2022

Find the bounce rate after the A/B test

The image shows an email inbox with a new message from Morgan Rockwell. The message body contains text about a bounce rate analysis comparing two landing pages. Below the message is a screenshot of a BI tool's 'Result Preview' section, which displays two tables: one for the first instance of /lander-1 and another for final analysis output comparing /home and /lander-1.

```

CREATE TEMPORARY TABLE first_landing_page
SELECT
    website_pageviews.website_session_id,
    MIN(website_pageview_id) as landing_page
FROM website_pageviews
LEFT JOIN website_sessions
ON website_pageviews.website_session_id = website_sessions.website_session_id
WHERE website_pageview_id >= (
    SELECT website_pageview_id FROM website_pageviews
    WHERE pageview_url = "/lander-1"
)
ORDER BY website_pageview_id
LIMIT 1
AND website_pageviews.created_at < '2012-07-28'
AND pageview_url IN ('/home', '/lander-1')
AND utm_campaign = 'nonbrand'
AND utm_source = 'gsearch'
GROUP BY 1;

CREATE TEMPORARY TABLE first_page_url
SELECT
    first_landing_page.website_session_id,

```

```

    website_pageviews.pageview_url
FROM first_landing_page
LEFT JOIN website_pageviews
ON first_landing_page.landing_page = website_pageviews.website_pageview_id;

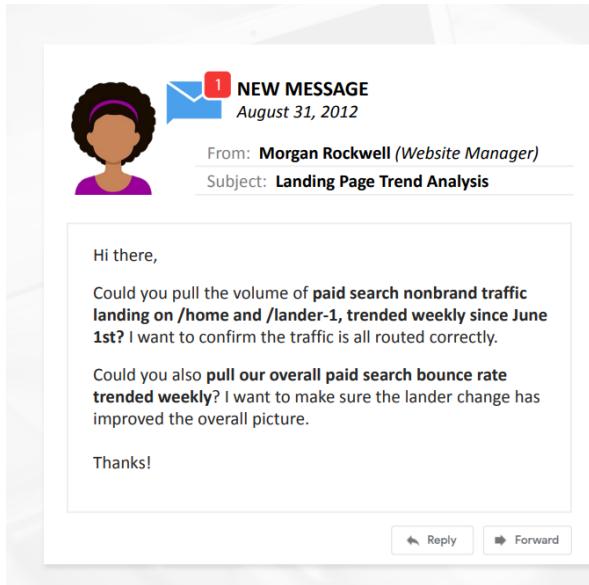
CREATE TEMPORARY TABLE bounced_sessions
SELECT
    first_page_url.website_session_id,
    first_page_url.pageview_url,
    COUNT(website_pageview_id) as Sessions
FROM first_page_url
LEFT JOIN website_pageviews
ON first_page_url.website_session_id = website_pageviews.website_session_id
GROUP BY 1,2
HAVING COUNT(website_pageview_id)=1;

SELECT
    first_page_url.pageview_url,
    COUNT(DISTINCT first_page_url.website_session_id) as Sessions,
    COUNT(DISTINCT bounced_sessions.website_session_id) as Bounce_sessions,
    COUNT(DISTINCT bounced_sessions.website_session_id)/COUNT(DISTINCT first_page_url.website_session_id) AS Bounce_rate
FROM first_page_url
LEFT JOIN bounced_sessions
ON first_page_url.website_session_id = bounced_sessions.website_session_id
GROUP BY 1;

```

Output

	pageview_url	Sessions	Bounce_sessions	Bounce_rate
▶	/home	2261	1319	0.5834
▶	/lander-1	2316	1233	0.5324



1 NEW MESSAGE
August 31, 2012

From: Morgan Rockwell (Website Manager)
Subject: Landing Page Trend Analysis

Hi there,

Could you pull the volume of **paid search nonbrand traffic landing on /home and /lander-1, trended weekly since June 1st?** I want to confirm the traffic is all routed correctly.

Could you also **pull our overall paid search bounce rate trended weekly?** I want to make sure the lander change has improved the overall picture.

Thanks!

Reply Forward

Result Preview

week_start_date	bounce_rate	home_sessions	lander_sessions
2012-06-01			
2012-06-03			
2012-06-10			
2012-06-17			
2012-06-24			
2012-07-01			
2012-07-08			
2012-07-15			
2012-07-22			
2012-07-29			
2012-08-05			
2012-08-12			
2012-08-19			
2012-08-26			

```

CREATE TEMPORARY TABLE first_pageview_id_table1
SELECT
    website_pageviews.website_session_id,
    MIN(website_pageviews.website_pageview_id) as first_pageview_id
FROM website_pageviews
LEFT JOIN website_sessions
ON website_sessions.website_session_id = website_pageviews.website_session_id
WHERE website_pageviews.created_at < '2012-08-31'
AND website_pageviews.created_at >= '2012-06-01'
AND utm_campaign = 'nonbrand'

```

```

AND utm_source = 'gsearch'
GROUP BY 1;

CREATE TEMPORARY TABLE landing_page_url3
SELECT
    first_pageview_id_table1.website_session_id,
    website_pageviews.pageview_url,
    website_pageviews.created_at
FROM first_pageview_id_table1
LEFT JOIN website_pageviews
ON first_pageview_id_table1.first_pageview_id = website_pageviews.website_pageview_id
GROUP BY 1;

CREATE TEMPORARY TABLE bounced_sessions_table3
SELECT
    landing_page_url3.website_session_id,
    website_pageviews.created_at,
    COUNT(website_pageviews.website_pageview_id) as Sessions
FROM landing_page_url3
LEFT JOIN website_pageviews
ON landing_page_url3.website_session_id = website_pageviews.website_session_id
GROUP BY 1
HAVING COUNT(website_pageviews.website_pageview_id) = 1;

SELECT
    MIN(DATE(landing_page_url3.created_at)) AS week_start_date,
    -- COUNT(DISTINCT landing_page_url3.website_session_id) AS Sessions,
    -- COUNT(DISTINCT bounced_sessions_table3.website_session_id) AS Bounced_sessions,
    COUNT(DISTINCT bounced_sessions_table3.website_session_id)/COUNT(DISTINCT landing_page_url3.website_session_id) AS Bounce_rate,
    COUNT(CASE WHEN pageview_url = '/home' THEN 1 ELSE NULL END) AS Home_sessions,
    COUNT(CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE NULL END) AS lander_sessions
FROM landing_page_url3
LEFT JOIN bounced_sessions_table3
ON landing_page_url3.website_session_id = bounced_sessions_table3.website_session_id
GROUP BY WEEK(landing_page_url3.created_at);

```

@September 12, 2022

Result Preview

Result Grid	Filter Rows:	Search	Export:
sessions_to_products to_mrffuzzy to_cart to_shipping to_billing to_thankyou	4493		

Result Grid	Filter Rows:	Search	Export:
lander_click_rt products_click_rt mrffuzzy_click_rt cart_click_rt shipping_click_rt billing_click_rt			

```

CREATE TEMPORARY TABLE funnel_sessions_details
SELECT
    website_sessions.website_session_id,
    website_pageviews.pageview_url,
    CASE WHEN pageview_url = '/lander-1' THEN 1 ELSE 0 END AS lander1_views,

```

```

CASE WHEN pageview_url = '/products' THEN 1 ELSE 0 END AS products_views,
CASE WHEN pageview_url = '/the-original-mr-fuzzy' THEN 1 ELSE 0 END AS mr_fuzzy_views,
CASE WHEN pageview_url = '/cart' THEN 1 ELSE 0 END AS cart_views,
CASE WHEN pageview_url = '/shipping' THEN 1 ELSE 0 END AS shipping_views,
CASE WHEN pageview_url = '/billing' THEN 1 ELSE 0 END AS billing_views,
CASE WHEN pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS thank_views
FROM website_sessions
LEFT JOIN website_pageviews
ON website_sessions.website_session_id = website_pageviews.website_session_id
WHERE website_sessions.created_at < '2012-09-05'
AND website_sessions.created_at > '2012-08-05'
AND website_sessions.utm_source='gsearch'
AND pageview_url IN ('/lander-1','/products','/the-original-mr-fuzzy', '/cart', '/shipping', '/billing', '/thank-you-for-your-order')
;

CREATE TEMPORARY TABLE funnel_by_website_session_id
SELECT
website_session_id,
MAX(lander1_views) as lander1,
MAX(products_views) as products,
MAX(mr_fuzzy_views) as mr_fuzzy,
MAX(cart_views) as cart,
MAX(shipping_views) as shipping,
MAX(billing_views) as billing,
MAX(thank_views) as thanks
FROM funnel_sessions_details
GROUP BY website_session_id;

SELECT
COUNT(DISTINCT CASE WHEN lander1=1 THEN website_session_id ELSE NULL END) AS sessions,
COUNT(DISTINCT CASE WHEN products=1 THEN website_session_id ELSE NULL END) AS to_products,
COUNT(DISTINCT CASE WHEN mr_fuzzy=1 THEN website_session_id ELSE NULL END) AS to_mrfuzzy,
COUNT(DISTINCT CASE WHEN cart=1 THEN website_session_id ELSE NULL END) AS to_cart,
COUNT(DISTINCT CASE WHEN shipping=1 THEN website_session_id ELSE NULL END) AS to_shipping,
COUNT(DISTINCT CASE WHEN billing=1 THEN website_session_id ELSE NULL END) AS to_billing,
COUNT(DISTINCT CASE WHEN thanks=1 THEN website_session_id ELSE NULL END) AS to_thankyou
FROM funnel_by_website_session_id;

SELECT
COUNT(DISTINCT CASE WHEN lander1=1 THEN website_session_id ELSE NULL END) AS sessions,
COUNT(DISTINCT CASE WHEN products=1 THEN website_session_id ELSE NULL END)
/COUNT(DISTINCT CASE WHEN lander1=1 THEN website_session_id ELSE NULL END) AS lander_click_rt,
COUNT(DISTINCT CASE WHEN mr_fuzzy=1 THEN website_session_id ELSE NULL END)
/COUNT(DISTINCT CASE WHEN products=1 THEN website_session_id ELSE NULL END) AS products_click_rt,
COUNT(DISTINCT CASE WHEN cart=1 THEN website_session_id ELSE NULL END)
/COUNT(DISTINCT CASE WHEN mr_fuzzy=1 THEN website_session_id ELSE NULL END) AS mrfuzzy_click_rt,
COUNT(DISTINCT CASE WHEN shipping=1 THEN website_session_id ELSE NULL END)
/COUNT(DISTINCT CASE WHEN cart=1 THEN website_session_id ELSE NULL END) AS cart_click_rt,
COUNT(DISTINCT CASE WHEN billing=1 THEN website_session_id ELSE NULL END)
/COUNT(DISTINCT CASE WHEN shipping=1 THEN website_session_id ELSE NULL END) AS shipping_click_rt,
COUNT(DISTINCT CASE WHEN thanks=1 THEN website_session_id ELSE NULL END)
/COUNT(DISTINCT CASE WHEN billing=1 THEN website_session_id ELSE NULL END) AS billing_click_rt
FROM funnel_by_website_session_id;

```

	sessions	to_products	to_mrfuzzy	to_cart	to_shipping	to_billing	to_thankyou
▶	4493	2115	1567	683	455	361	158

	sessions	lander_click_rt	products_click_rt	mrfuzzy_click_rt	cart_click_rt	shipping_click_rt	billing_click_rt
▶	4493	0.4707	0.7409	0.4359	0.6662	0.7934	0.4377

@September 14, 2022

NEW MESSAGE
November 10, 2012

From: Morgan Rockwell (Website Manager)
Subject: Conversion Funnel Test Results

Hello!

We tested an updated billing page based on your funnel analysis. Can you take a look and see whether /billing-2 is doing any better than the original /billing page?

We're wondering what % of sessions on those pages end up placing an order. FYI – we ran this test for all traffic, not just for our search visitors.

Thanks!
-Morgan

[Reply](#) [Forward](#)

Result Preview

-- finding the first time /billing-2 was seen

first_created_at	first_pv_id
2012-11-10 10:00:00	1234567890

-- final test analysis output

billing_version_seen	sessions	orders	billing_to_order_rt
/billing	657	300	0.4566
/billing-2	654	410	0.6269

```

SELECT
    pageview_url,
    COUNT(DISTINCT website_session_id) as Sessions,
    COUNT(DISTINCT order_id) as Orders,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT website_session_id) AS Sessions_to_order_rt
FROM
(SELECT
    website_pageviews.website_session_id,
    pageview_url,
    order_id
FROM website_pageviews
LEFT JOIN orders
    ON website_pageviews.website_session_id = orders.website_session_id
WHERE website_pageview_id >= (SELECT website_pageview_id FROM website_pageviews
WHERE pageview_url = '/billing-2'
ORDER BY website_pageview_id
LIMIT 1)
AND pageview_url IN ('/billing','/billing-2')
AND website_pageviews.created_at < '2012-11-10'
) AS web_order
GROUP BY 1;

```

	pageview_url	Sessions	Orders	Sessions_to_order_rt
▶	/billing	657	300	0.4566
	/billing-2	654	410	0.6269

@September 19, 2022

Mid - Course Project

@October 4, 2022

 **NEW MESSAGE**
November 29, 2012

From: Tom Parmesan (Marketing Director)
Subject: Expanded Channel Portfolio

Hi there,

With gsearch doing well and the site performing better, we launched a second paid search channel, bsearch, around August 22.

Can you pull weekly trended session volume since then and compare to gsearch nonbrand so I can get a sense for how important this will be for the business?

Thanks, Tom

[Reply](#) [Forward](#)

Result Preview

Result Grid Filter Rows: Search

week_start_date	gsearch_sessions	bsearch_sessions
2012-08-22		
2012-08-26		
2012-09-02		
2012-09-09		
2012-09-16		
2012-09-23		
2012-09-30		
2012-10-07		
2012-10-14		
2012-10-21		
2012-10-28		
2012-11-04		
2012-11-11		
2012-11-18		
2012-11-25		

```

SELECT
    MIN(DATE(created_at)) as Week_start_date,
    COUNT(CASE WHEN utm_source='gsearch' THEN website_session_id ELSE NULL END) AS gsearch_session,
    COUNT(CASE WHEN utm_source='bsearch' THEN website_session_id ELSE NULL END) AS bsearch_session
FROM website_sessions
WHERE utm_source IN ('gsearch', 'bsearch')
AND created_at < '2012-11-29'
AND created_at >= '2012-08-22'
GROUP BY WEEK(created_at);

```

	Week_start_date	gsearch_session	bsearch_session
▶	2012-08-22	609	201
	2012-08-26	1109	361
	2012-09-02	980	307
	2012-09-09	1006	343
	2012-09-16	1230	387
	2012-09-23	1116	341
	2012-09-30	1058	330
	2012-10-07	1085	351
	2012-10-14	1335	439
	2012-10-21	1375	455
	2012-10-28	1282	410
	2012-11-04	1427	446
	2012-11-11	1317	458
	2012-11-18	3654	1119
	2012-11-25	2390	792

 NEW MESSAGE
November 30, 2012

From: Tom Parmesan (Marketing Director)
Subject: Comparing Our Channels

Hi there,

I'd like to learn more about the **bsearch nonbrand** campaign. Could you please pull the **percentage of traffic coming on Mobile**, and **compare that to gsearch**?

Feel free to dig around and share anything else you find interesting. **Aggregate data since August 22nd** is great, no need to show trending at this point.

Thanks, Tom

[Reply](#) [Forward](#)

Result Preview

Result Grid   Filter Rows: Search

utm_source	sessions	mobile_sessions	pct_mobile
bsearch	6522	562	0.0862
gsearch	20073	4921	0.2452

```

SELECT
    utm_source,
    COUNT(DISTINCT website_session_id) as Sessions,
    COUNT(CASE WHEN device_type='mobile' THEN website_session_id ELSE NULL END) AS mob_sessions,
    COUNT(CASE WHEN device_type='mobile' THEN website_session_id ELSE NULL END)/
    COUNT(DISTINCT website_session_id) AS pct_mob
FROM website_sessions
WHERE utm_source IN ('gsearch','bsearch')
AND created_at < '2012-11-30'
AND created_at > '2012-08-22'
AND utm_campaign = 'nonbrand'
GROUP BY 1;
  
```

	utm_source	Sessions	mob_sessions	pct_mob
▶	bsearch	6522	562	0.0862
	gsearch	20073	4921	0.2452

NEW MESSAGE
December 01, 2012

From: Tom Parmesan (Marketing Director)
Subject: Multi-Channel Bidding

Hi there,
I'm wondering if bsearch nonbrand should have the same bids as gsearch. Could you pull **nonbrand conversion rates from session to order for gsearch and bsearch, and slice the data by device type?**
Please analyze data from **August 22 to September 18**; we ran a special pre-holiday campaign for gsearch starting on **September 19th**, so the data after that isn't fair game.
Thanks, Tom

Reply Forward

Result Preview

Result Grid Filter Rows: Search

device_type	utm_source	sessions	orders	conv_rate
desktop	bsearch	1162	44	0.0379
desktop	gsearch	3011	136	0.0452
mobile	bsearch	130	1	0.0077
mobile	gsearch	1015	13	0.0128

```

SELECT
    device_type,
    utm_source,
    COUNT(DISTINCT website_sessions.website_session_id) AS Sessions,
    COUNT(DISTINCT order_id) as Orders,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS cvt
FROM website_sessions
LEFT JOIN orders
    ON website_sessions.website_session_id = orders.website_session_id
WHERE utm_source IN ('gsearch','bsearch')
AND website_sessions.created_at < '2012-09-19'
AND website_sessions.created_at > '2012-08-22'
AND utm_campaign = 'nonbrand'
GROUP BY 1,2;

```

	device_type	utm_source	Sessions	Orders	cvt
▶	desktop	bsearch	1162	44	0.0379
	desktop	gsearch	3011	136	0.0452
	mobile	bsearch	130	1	0.0077
	mobile	gsearch	1015	13	0.0128



NEW MESSAGE
December 22, 2012

From: Tom Parmesan (Marketing Director)
Subject: Impact of Bid Changes

Hi there,
Based on your last analysis, we bid down bsearch nonbrand on December 2nd.
Can you pull weekly session volume for gsearch and bsearch nonbrand, broken down by device, since November 4th?
If you can include a comparison metric to show bsearch as a percent of gsearch for each device, that would be great too.
Thanks, Tom

[Reply](#) [Forward](#)

Result Preview

week_start_date	g_dtop_sessions	b_dtop_sessions	b_top_pct_of_g_top	g_mob_sessions	b_mob_sessions	b_mob_pct_g_mob
2012-11-04	1027	400	0.3895	323	29	0.0898
2012-11-11	956	401	0.4195	290	37	0.1276
2012-11-18	2655	1008	0.3797	853	85	0.0996
2012-11-25	2058	843	0.4096	692	62	0.0896
2012-12-02	1326	517	0.3899	396	31	0.0783
2012-12-09	1277	293	0.2294	424	46	0.1085
2012-12-16	1270	348	0.2740	376	41	0.1090

```

SELECT
    MIN(DATE(created_at)) week_start_date,
    COUNT(CASE WHEN device_type = 'desktop' AND utm_source = 'gsearch' THEN website_session_id ELSE NULL END) AS g_dtop_sessions,
    COUNT(CASE WHEN device_type = 'desktop' AND utm_source = 'bsearch' THEN website_session_id ELSE NULL END) AS b_dtop_sessions,
    COUNT(CASE WHEN device_type = 'desktop' AND utm_source = 'bsearch' THEN website_session_id ELSE NULL END)/
    COUNT(CASE WHEN device_type = 'desktop' AND utm_source = 'gsearch' THEN website_session_id ELSE NULL END) AS b_top_pct_of_g_top,
    COUNT(CASE WHEN device_type = 'mobile' AND utm_source = 'gsearch' THEN website_session_id ELSE NULL END) AS g_mob_sessions,
    COUNT(CASE WHEN device_type = 'mobile' AND utm_source = 'bsearch' THEN website_session_id ELSE NULL END) AS b_mob_sessions,
    COUNT(CASE WHEN device_type = 'mobile' AND utm_source = 'bsearch' THEN website_session_id ELSE NULL END)/
    COUNT(CASE WHEN device_type = 'mobile' AND utm_source = 'gsearch' THEN website_session_id ELSE NULL END) AS b_mob_pct_g_mob
FROM website_sessions
WHERE utm_source IN ('gsearch', 'bsearch')
AND website_sessions.created_at > '2012-11-04'
AND website_sessions.created_at < '2012-12-22'
AND utm_campaign = 'nonbrand'
GROUP BY WEEK(created_at);

```

	week_start_date	g_dtop_sessions	b_dtop_sessions	b_top_pct_of_g_top	g_mob_sessions	b_mob_sessions	b_mob_pct_g_mob
▶	2012-11-04	1027	400	0.3895	323	29	0.0898
	2012-11-11	956	401	0.4195	290	37	0.1276
	2012-11-18	2655	1008	0.3797	853	85	0.0996
	2012-11-25	2058	843	0.4096	692	62	0.0896
	2012-12-02	1326	517	0.3899	396	31	0.0783
	2012-12-09	1277	293	0.2294	424	46	0.1085
	2012-12-16	1270	348	0.2740	376	41	0.1090

1 NEW MESSAGE
December 23, 2012

From: Cindy Sharp (CEO)
Subject: Site traffic breakdown

Good morning,
A potential investor is asking if we're building any momentum with our brand or if we'll need to keep relying on paid traffic.
Could you pull organic search, direct type in, and paid brand search sessions by month, and show those sessions as a % of paid search nonbrand?
-Cindy

[Reply](#) [Forward](#)

Result Preview

yr	mo	nonbrand	brand	brand_pct_of_nonbrand	direct	direct_pct_of_nonbrand	organic	organic_pct_of_nonbrand
2012	3	1852	10	0.0054	9	0.0049	8	0.0043
2012	4	3509	76	0.0217	71	0.0202	78	0.0222
2012	5	3295	140	0.0425	151	0.0458	150	0.0455
2012	6	3439	164	0.0477	170	0.0494	190	0.0552
2012	7	3660	195	0.0533	187	0.0511	207	0.0566
2012	8	5318	264	0.0496	250	0.0470	265	0.0498
2012	9	5591	339	0.0606	285	0.0510	331	0.0592
2012	10	6883	432	0.0628	440	0.0639	428	0.0622
2012	11	12260	556	0.0454	571	0.0466	624	0.0509
2012	12	6643	464	0.0698	482	0.0726	492	0.0741

```

SELECT
    YEAR(created_at) as Yr,
    Month(created_at) as mo,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN website_session_id ELSE NULL END) AS non_brand,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN website_session_id ELSE NULL END) AS brand,
    COUNT(DISTINCT CASE WHEN utm_campaign = 'brand' THEN website_session_id ELSE NULL END)/
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN website_session_id ELSE NULL END) AS brand_pct_of_non_brand,
    COUNT(DISTINCT CASE WHEN http_referer IS NULL THEN website_session_id ELSE NULL END) AS Direct,
    COUNT(DISTINCT CASE WHEN http_referer IS NULL THEN website_session_id ELSE NULL END)/
    COUNT(DISTINCT CASE WHEN utm_campaign = 'nonbrand' THEN website_session_id ELSE NULL END) AS direct_pct_of_non_brand,
    COUNT(DISTINCT CASE WHEN http_referer IN ('https://www.gsearch.com', 'https://www.bsearch.com') AND utm_source IS NULL THEN website_session_id ELSE NULL END) AS Organic,
    COUNT(DISTINCT CASE WHEN http_referer IN ('https://www.gsearch.com', 'https://www.bsearch.com') AND utm_source IS NULL THEN website_session_id ELSE NULL END) AS organic_pct_of_non_brand
FROM website_sessions
WHERE website_sessions.created_at < '2012-12-23'
GROUP BY 1, 2;

```

	Yr	mo	non_brand	brand	brand_pct_of_non_brand	Direct	direct_pct_of_non_brand	Orgnaic	organic_pct_of_non_brand
▶	2012	3	1852	10	0.0054	9	0.0049	8	0.0043
	2012	4	3509	76	0.0217	71	0.0202	78	0.0222
	2012	5	3295	140	0.0425	151	0.0458	150	0.0455
	2012	6	3439	164	0.0477	170	0.0494	190	0.0552
	2012	7	3660	195	0.0533	187	0.0511	207	0.0566
	2012	8	5318	264	0.0496	250	0.0470	265	0.0498
	2012	9	5591	339	0.0606	285	0.0510	331	0.0592
	2012	10	6883	432	0.0628	440	0.0639	428	0.0622
	2012	11	12260	556	0.0454	571	0.0466	624	0.0509
	2012	12	6643	464	0.0698	482	0.0726	492	0.0741

@October 5, 2022

1 NEW MESSAGE
January 02, 2013

From: Cindy Sharp (CEO)
Subject: Understanding Seasonality

Good morning,

2012 was a great year for us. As we continue to grow, we should take a look at 2012's monthly and weekly volume patterns, to see if we can find any seasonal trends we should plan for in 2013.

If you can pull session volume and order volume, that would be excellent.

Thanks,
-Cindy

Reply Forward

Result Preview

week_start_date	sessions	orders
2012-01-01	10	10
2012-01-08	10	10
2012-01-15	10	10
2012-01-22	10	10
2012-01-29	10	10
2012-02-05	10	10
2012-02-12	10	10
2012-02-19	10	10
2012-02-26	10	10
2012-03-05	10	10
2012-03-12	10	10
2012-03-19	10	10
2012-03-26	10	10
2012-04-02	10	10
2012-04-09	10	10
2012-04-16	10	10
2012-04-23	10	10
2012-04-30	10	10
2012-05-07	10	10
2012-05-14	10	10
2012-05-21	10	10
2012-05-28	10	10
2012-06-04	10	10
2012-06-11	10	10
2012-06-18	10	10
2012-06-25	10	10
2012-07-02	10	10
2012-07-09	10	10
2012-07-16	10	10
2012-07-23	10	10
2012-07-30	10	10
2012-08-06	10	10
2012-08-13	10	10
2012-08-20	10	10
2012-08-27	10	10
2012-09-03	10	10
2012-09-10	10	10
2012-09-17	10	10
2012-09-24	10	10
2012-10-01	10	10
2012-10-08	10	10
2012-10-15	10	10
2012-10-22	10	10
2012-10-29	10	10
2012-11-05	10	10
2012-11-12	10	10
2012-11-19	10	10
2012-11-26	10	10
2012-12-03	10	10
2012-12-10	10	10
2012-12-17	10	10
2012-12-24	10	10
2012-12-31	10	10

```
SELECT
    YEAR(website_sessions.created_at) AS yr,
    MONTH(website_sessions.created_at) AS Mo,
    COUNT(DISTINCT website_sessions.website_session_id) as Sessions,
    COUNT(DISTINCT order_id) as Orders
FROM website_sessions
LEFT JOIN orders
    ON website_sessions.website_session_id = orders.website_session_id
WHERE website_sessions.created_at < '2013-01-02'
GROUP BY 1,2;
```

	yr	Mo	Sessions	Orders
▶	2012	3	1879	60
	2012	4	3734	99
	2012	5	3736	108
	2012	6	3963	140
	2012	7	4249	169
	2012	8	6097	228
	2012	9	6546	287
	2012	10	8183	371
	2012	11	14011	618
	2012	12	10072	506
	2013	1	217	13

```
SELECT
    MIN(DATE(website_sessions.created_at)) AS Date,
    COUNT(DISTINCT website_sessions.website_session_id) as Sessions,
    COUNT(DISTINCT order_id) as Orders
FROM website_sessions
LEFT JOIN orders
    ON website_sessions.website_session_id = orders.website_session_id
WHERE website_sessions.created_at < '2013-01-02'
GROUP BY WEEK(website_sessions.created_at);
```

	Date	Sessions	Orders
▶	2012-03-19	896	25
	2012-03-25	983	35
	2012-04-01	1193	29
	2012-04-08	1029	28
	2012-04-15	679	22
	2012-04-22	655	18
	2012-04-29	770	19
	2012-05-06	798	17
	2012-05-13	706	23
	2012-05-20	965	28
	2012-05-27	875	31
	2012-06-03	920	34
	2012-06-10	994	29
	2012-06-17	966	37
	2012-06-24	883	32
	2012-07-01	892	30
	2012-07-08	925	36
	2012-07-15	987	47

1 NEW MESSAGE
January 05, 2013

From: Cindy Sharp (CEO)
Subject: Data for Customer Service

Good morning,
We're considering adding live chat support to the website to improve our customer experience. Could you analyze the **average website session volume, by hour of day and by day week**, so that we can staff appropriately?

Let's avoid the holiday time period and use a date range of **Sep 15 - Nov 15, 2013**.

Thanks, Cindy

Reply Forward

Result Preview

Result Grid							Filter Rows:	Q Search
hr	mon	tue	wed	thu	fri	sat	sun	
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```

SELECT
    HOUR(created_at) as Hr,
    COUNT(CASE WHEN WEEKDAY(created_at)=0 THEN website_session_id ELSE NULL END) AS Mo,
    COUNT(CASE WHEN WEEKDAY(created_at)=1 THEN website_session_id ELSE NULL END) AS Tue,
    COUNT(CASE WHEN WEEKDAY(created_at)=2 THEN website_session_id ELSE NULL END) AS Wed,
    COUNT(CASE WHEN WEEKDAY(created_at)=3 THEN website_session_id ELSE NULL END) AS Thu,
    COUNT(CASE WHEN WEEKDAY(created_at)=4 THEN website_session_id ELSE NULL END) AS Fri,
    COUNT(CASE WHEN WEEKDAY(created_at)=5 THEN website_session_id ELSE NULL END) AS Sat,
    COUNT(CASE WHEN WEEKDAY(created_at)=6 THEN website_session_id ELSE NULL END) AS Sun
FROM website_sessions
WHERE created_at < '2012-11-15'
AND created_at > '2012-09-15'
GROUP BY 1
ORDER BY 1;

```

	Hr	Mo	Tue	Wed	Thu	Fri	Sat	Sun
0	78	69	57	59	54	45	45	
1	59	60	48	39	57	45	24	
2	55	40	40	49	37	33	24	
3	51	36	42	32	29	35	27	
4	47	57	54	32	43	22	22	
5	45	49	46	43	37	39	35	
6	49	50	43	48	54	36	23	
7	66	70	67	85	56	51	43	
8	111	110	117	132	84	34	37	
9	158	141	176	154	140	68	54	
10	166	159	189	147	152	75	57	
11	162	172	224	173	167	65	69	
12	190	210	205	193	152	77	55	
13	160	207	187	165	173	73	76	

	14	161	194	201	148	156	78	60
▶	15	194	154	228	188	170	62	64
	16	190	213	213	157	167	61	59
	17	175	143	182	158	103	58	68
	18	114	135	133	122	87	48	61
	19	112	127	120	93	114	64	58
	20	109	112	128	85	82	51	76
	21	82	113	103	75	58	51	92
	22	82	90	88	97	48	51	92
	23	79	77	86	85	61	48	75

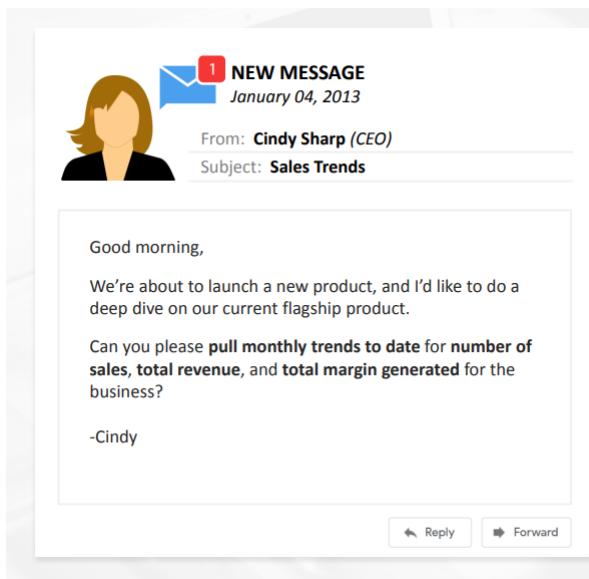
```

SELECT
Hr,
ROUND(AVG(CASE WHEN Wekdy=0 THEN Sessions ELSE NULL END),1) AS Mon,
ROUND(AVG(CASE WHEN Wekdy=1 THEN Sessions ELSE NULL END),1) AS Tue,
ROUND(AVG(CASE WHEN Wekdy=2 THEN Sessions ELSE NULL END),1) AS Wed,
ROUND(AVG(CASE WHEN Wekdy=3 THEN Sessions ELSE NULL END),1) AS Thr,
ROUND(AVG(CASE WHEN Wekdy=4 THEN Sessions ELSE NULL END),1) AS Fri,
ROUND(AVG(CASE WHEN Wekdy=5 THEN Sessions ELSE NULL END),1) AS Sat,
ROUND(AVG(CASE WHEN Wekdy=6 THEN Sessions ELSE NULL END),1) AS Sun
FROM
(
SELECT
DATE(created_at) as Date,
WEEKDAY(created_at) AS Wekdy,
HOUR(created_at) AS Hr,
COUNT( DISTINCT website_session_id) AS Sessions
FROM website_sessions
WHERE created_at BETWEEN '2012-09-15' AND '2012-11-15'
GROUP BY 1,2,3
) AS hourly_daily_sessions
GROUP BY 1
ORDER BY 1;

```

	Hr	Mon	Tue	Wed	Thr	Fri	Sat	Sun
▶	0	8.7	7.7	6.3	7.4	6.8	5.0	5.0
	1	6.6	6.7	5.3	4.9	7.1	5.0	3.0
	2	6.1	4.4	4.4	6.1	4.6	3.7	3.0
	3	5.7	4.0	4.7	4.6	3.6	3.9	3.4
	4	5.9	6.3	6.0	4.0	6.1	2.8	2.4
	5	5.0	5.4	5.1	5.4	4.6	4.3	3.9
	6	5.4	5.6	4.8	6.0	6.8	4.0	2.6
	7	7.3	7.8	7.4	10.6	7.0	5.7	4.8
	8	12.3	12.2	13.0	16.5	10.5	4.3	4.1
	9	17.6	15.7	19.6	19.3	17.5	7.6	6.0
	10	18.4	17.7	21.0	18.4	19.0	8.3	6.3
	11	18.0	19.1	24.9	21.6	20.9	7.2	7.7
	12	21.1	23.3	22.8	24.1	19.0	8.6	6.1

--	--	--	--	--	--	--	--	--
13	17.8	23.0	20.8	20.6	21.6	8.1	8.4	
14	17.9	21.6	22.3	18.5	19.5	8.7	6.7	
15	21.6	17.1	25.3	23.5	21.3	6.9	7.1	
16	21.1	23.7	23.7	19.6	20.9	7.6	6.6	
17	19.4	15.9	20.2	19.8	12.9	6.4	7.6	
18	12.7	15.0	14.8	15.3	10.9	5.3	6.8	
19	12.4	14.1	13.3	11.6	14.3	7.1	6.4	
20	12.1	12.4	14.2	10.6	10.3	5.7	8.4	
21	9.1	12.6	11.4	9.4	7.3	5.7	10.2	
22	9.1	10.0	9.8	12.1	6.0	5.7	10.2	
23	8.8	8.6	9.6	10.6	7.6	5.3	8.3	



1 NEW MESSAGE
January 04, 2013

From: Cindy Sharp (CEO)
Subject: Sales Trends

Good morning,
We're about to launch a new product, and I'd like to do a deep dive on our current flagship product.
Can you please pull monthly trends to date for number of sales, total revenue, and total margin generated for the business?
-Cindy

[Reply](#) [Forward](#)

Result Preview

yr	mo	number_of_sales	total_revenue	total_margin
2012	3			
2012	4			
2012	5			
2012	6			
2012	7			
2012	8			
2012	9			
2012	10			
2012	11			
2012	12			
2013	1			

```

SELECT
    YEAR(created_at) AS Yr,
    MONTH(created_at) AS Mo,
    COUNT(DISTINCT order_id) as num_of_sales,
    SUM(price_usd) AS Total_revenue,
    SUM(price_usd - cogs_usd) AS Total_Margin
FROM orders
WHERE created_at < '2013-01-04'
GROUP BY 1,2;

```

	Yr	Mo	num_of_sales	Total_revenue	Total_Margin
▶	2012	3	60	2999.40	1830.00
	2012	4	99	4949.01	3019.50
	2012	5	108	5398.92	3294.00
	2012	6	140	6998.60	4270.00
	2012	7	169	8448.31	5154.50
	2012	8	228	11397.72	6954.00
	2012	9	287	14347.13	8753.50
	2012	10	371	18546.29	11315.50
	2012	11	618	30893.82	18849.00
	2012	12	506	25294.94	15433.00
	2013	1	42	2099.58	1281.00

1 NEW MESSAGE
April 05, 2013

From: Cindy Sharp (CEO)
Subject: Impact of New Product Launch

Good morning,
We launched our second product back on January 6th. Can you pull together some trended analysis?
I'd like to see **monthly order volume, overall conversion rates, revenue per session, and a breakdown of sales by product**, all for the time period **since April 1, 2013**.

Thanks,
-Cindy

Reply Forward

Result Preview

yr	mo	orders	conv_rate	revenue_per_session	product_one_orders	product_two_orders
2012	4	99	0.0265	1.325391	99	0
2012	5	108	0.0289	1.445107	108	0
2012	6	140	0.0353	1.765985	140	0
2012	7	169	0.0398	1.988305	169	0
2012	8	228	0.0374	1.869398	228	0
2012	9	287	0.0438	2.191740	287	0
2012	10	371	0.0453	2.266441	371	0
2012	11	618	0.0441	2.204969	618	0
2012	12	506	0.0502	2.511412	506	0
2013	1	391	0.0611	3.127025	344	47
2013	2	497	0.0693	3.692108	335	162
2013	3	385	0.0615	3.176269	320	65

```

SELECT
    YEAR(website_sessions.created_at) AS Yr,
    MONTH(website_sessions.created_at) AS Mo,
    COUNT(DISTINCT order_id) AS Orders,
    COUNT(DISTINCT order_id)/
    COUNT(DISTINCT website_sessions.website_session_id) AS Cov_rat,
    SUM(price_usd)/ COUNT(DISTINCT website_sessions.website_session_id) AS Revenue_per_session,
    COUNT(DISTINCT CASE WHEN primary_product_id = 1 THEN order_id ELSE NULL END) AS Product_one_orders,
    COUNT(DISTINCT CASE WHEN primary_product_id = 2 THEN order_id ELSE NULL END) AS Product_two_orders
FROM website_sessions
LEFT JOIN orders
    ON website_sessions.website_session_id = orders.website_session_id
WHERE website_sessions.created_at BETWEEN '2012-04-01' AND '2013-04-01'
GROUP BY 1,2;

```

	Yr	Mo	Orders	Cov_rat	Revenue_per_session	Product_one_orders	Product_two_orders
▶	2012	4	99	0.0265	1.325391	99	0
	2012	5	108	0.0289	1.445107	108	0
	2012	6	140	0.0353	1.765985	140	0
	2012	7	169	0.0398	1.988305	169	0
	2012	8	228	0.0374	1.869398	228	0
	2012	9	287	0.0438	2.191740	287	0
	2012	10	371	0.0453	2.266441	371	0
	2012	11	618	0.0441	2.204969	618	0
	2012	12	506	0.0502	2.511412	506	0
	2013	1	391	0.0611	3.127025	344	47
	2013	2	497	0.0693	3.692108	335	162
	2013	3	385	0.0615	3.176269	320	65

@October 7, 2022

NEW MESSAGE
April 06, 2014

From: Morgan Rockwell (Website Manager)
Subject: Help w/ User Pathing

Hi there!

Now that we have a new product, I'm thinking about our user path and conversion funnel. Let's look at sessions which hit the /products page and see where they went next.

Could you please pull clickthrough rates from /products since the new product launch on January 6th 2013, by product, and compare to the 3 months leading up to launch as a baseline?

Thanks, Morgan

Reply Forward

Result Preview

time_period	sessions_w_next_pg	pct_w_next_pg	to_mrfuzzy	pct_to_mrfuzzy	to_lovebear	pct_to_lovebear
A.Pre_Product_2						
B.Post_Product_2						

```

CREATE TEMPORARY TABLE Pageview_table
SELECT
    website_session_id,
    website_pageview_id,
    created_at,
    CASE
        WHEN created_at < '2013-01-06' THEN 'A. Pre_Product_2'
        WHEN created_at >= '2013-01-06' THEN 'B. Post_Product_2'
        ELSE 'Uh_Check the date range'
    END AS time_period
FROM website_pageviews
WHERE created_at < '2013-04-06'
AND created_at > '2012-10-06'
AND pageview_url = '/products';

CREATE TEMPORARY TABLE sessions_w_next_pageview_id
SELECT
    Pageview_table.time_period,
    Pageview_table.website_session_id,
    MIN(website_pageviews.website_pageview_id) AS Min_pageview_id
FROM Pageview_table
LEFT JOIN website_pageviews
    ON Pageview_table.website_session_id = website_pageviews.website_session_id
        AND website_pageviews.website_pageview_id > Pageview_table.website_pageview_id
GROUP BY 1,2;

CREATE TEMPORARY TABLE sessions_w_next_page_url
SELECT
    sessions_w_next_pageview_id.time_period,
    sessions_w_next_pageview_id.website_session_id,
    website_pageviews.pageview_url AS next_page_url
FROM sessions_w_next_pageview_id
LEFT JOIN website_pageviews
    ON sessions_w_next_pageview_id.Min_pageview_id = website_pageviews.website_pageview_id;

SELECT
    time_period,
    COUNT(DISTINCT website_session_id) as Sessions,
    COUNT(DISTINCT CASE WHEN next_page_url IS NOT NULL THEN website_session_id ELSE NULL END) AS w_next_page,
    COUNT(DISTINCT CASE WHEN next_page_url IS NOT NULL THEN website_session_id ELSE NULL END)/
    COUNT(DISTINCT website_session_id) AS pct_w_next_page_sessions,
    COUNT(DISTINCT CASE WHEN next_page_url = '/the-original-mr-fuzzy' THEN website_session_id ELSE NULL END) AS to_mrfuzzy,
    COUNT(DISTINCT CASE WHEN next_page_url = '/the-original-mr-fuzzy' THEN website_session_id ELSE NULL END)/
    COUNT(DISTINCT website_session_id) AS pct_to_mrfuzzy,
    COUNT(DISTINCT CASE WHEN next_page_url = '/the-forever-love-bear' THEN website_session_id ELSE NULL END) AS to_lovebear,
    COUNT(DISTINCT CASE WHEN next_page_url = '/the-forever-love-bear' THEN website_session_id ELSE NULL END)/
    COUNT(DISTINCT website_session_id) AS pct_to_lovebear
FROM sessions_w_next_page_url
GROUP BY 1;

```

	time_period	Sessions	w_next_page	pct_w_next_page_sessions	to_mr_fuzzy	pct_to_mr_fuzzy	to_lovebear	pct_to_lovebear
▶	A. Pre_Product_2	15696	11347	0.7229	11347	0.7229	0	0.0000
	B. Post_Product_2	10709	8200	0.7657	6654	0.6213	1546	0.1444

@October 8, 2022

Result Preview

product_seen	sessions	to_cart	to_shipping	to_billing	to_thankyou
▶ lovebear					
mrfuzzy					

product_seen	product_page_click_rt	cart_click_rt	shipping_click_rt	billing_click_rt
lovebear	0.00000	0.00000	0.00000	0.00000
mrfuzzy	0.00000	0.00000	0.00000	0.00000

```

CREATE TEMPORARY TABLE product_w_pageview_id
SELECT
    website_session_id,
    website_pageview_id,
    CASE
        WHEN pageview_url = '/the-original-mr-fuzzy' THEN 'mrfuzzy'
        WHEN pageview_url = '/the-forever-love-bear' THEN 'lovebear'
        ELSE 'NA'
    END AS product_seen
FROM website_pageviews
WHERE created_at < '2013-04-10'
AND created_at >= '2013-01-06'
AND pageview_url IN ('/the-original-mr-fuzzy', '/the-forever-love-bear');

CREATE TEMPORARY TABLE products_page_funnel_first_stage
SELECT
    product_w_pageview_id.product_seen,
    product_w_pageview_id.website_session_id,
    website_pageviews.pageview_url,
    CASE WHEN website_pageviews.pageview_url = '/cart' THEN 1 ELSE 0 END AS cart_view,
    CASE WHEN website_pageviews.pageview_url = '/shipping' THEN 1 ELSE 0 END AS shipping_view,
    CASE WHEN website_pageviews.pageview_url = '/billing-2' THEN 1 ELSE 0 END AS billing_view,
    CASE WHEN website_pageviews.pageview_url = '/thank-you-for-your-order' THEN 1 ELSE 0 END AS thank_view
FROM product_w_pageview_id
LEFT JOIN website_pageviews
ON product_w_pageview_id.website_session_id = website_pageviews.website_session_id
    AND website_pageviews.website_pageview_id > product_w_pageview_id.website_pageview_id ;

CREATE TEMPORARY TABLE products_page_funnel_second_stage
SELECT
    products_page_funnel_first_stage.product_seen,
    products_page_funnel_first_stage.website_session_id,
    MAX(cart_view) AS Cart,
    MAX(shipping_view) AS Shipping,
    MAX(billing_view) AS Billing,
    MAX(thank_view) AS thank_you
FROM products_page_funnel_first_stage
GROUP BY 2;

```

```

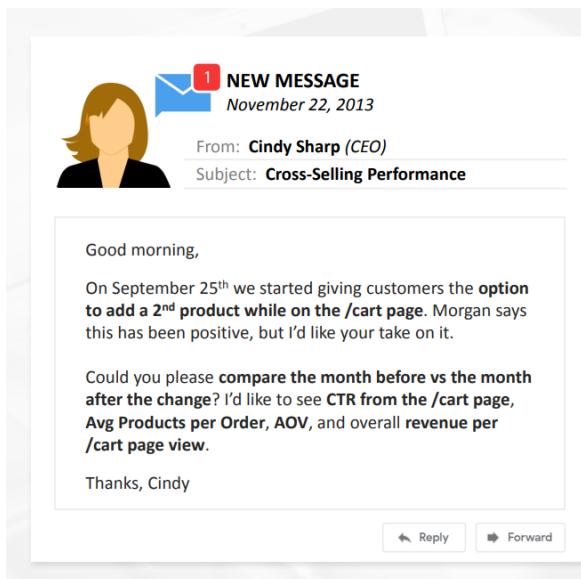
SELECT
products_page_funnel_second_stage.product_seen,
COUNT(DISTINCT website_session_id) AS Sessions,
COUNT(DISTINCT CASE WHEN Cart=1 THEN website_session_id ELSE NULL END) AS to_cart,
COUNT(DISTINCT CASE WHEN Shipping=1 THEN website_session_id ELSE NULL END) AS to_shipping,
COUNT(DISTINCT CASE WHEN billing=1 THEN website_session_id ELSE NULL END) AS to_billing,
COUNT(DISTINCT CASE WHEN thank_you=1 THEN website_session_id ELSE NULL END) AS to_thankyou
FROM products_page_funnel_second_stage
GROUP BY 1;

SELECT
products_page_funnel_second_stage.product_seen,
COUNT(DISTINCT CASE WHEN Cart=1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT website_session_id) AS product_to_cart_clk_rt,
COUNT(DISTINCT CASE WHEN Shipping=1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN Cart=1 THEN website_session_id ELSE NULL END) AS cart_to_shipping_clk_rt,
COUNT(DISTINCT CASE WHEN billing=1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN Shipping=1 THEN website_session_id ELSE NULL END) AS shipping_to_billing_clk_rt,
COUNT(DISTINCT CASE WHEN thank_you=1 THEN website_session_id ELSE NULL END)/COUNT(DISTINCT CASE WHEN billing=1 THEN website_session_id ELSE NULL END) AS billing_to_thankyou_clk_rt
FROM products_page_funnel_second_stage
GROUP BY 1;

```

	product_seen	Sessions	to_cart	to_shipping	to_billing	to_thankyou
▶	lovebear	1599	877	603	488	301
	mrfuzzy	6985	3038	2084	1710	1088

	product_seen	product_to_cart_clk_rt	cart_to_shipping_clk_rt	shipping_to_billing_clk_rt	billing_to_thankyou_clk_rt
▶	lovebear	0.5485	0.6876	0.8093	0.6168
	mrfuzzy	0.4349	0.6860	0.8205	0.6363



Result Preview

Result Grid	Filter Rows:	Search	Export:			
time_period	cart_sessions	clickthroughs	cart_ctr	products_per_order	aov	rev_per_cart_session
▶ A. Pre_Cross_Sell						
B. Post_Cross_Sell						

```

CREATE TEMPORARY TABLE cart_sessions
SELECT
website_session_id,
website_pageview_id,
CASE
WHEN created_at < '2013-09-25' THEN 'A. Pre_Cross_sell'
WHEN created_at >= '2013-09-25' THEN 'B. Post_cross_sell'
ELSE 'range is wrong'
END AS 'time_period'
FROM website_pageviews
WHERE created_at > '2013-08-25' -- before one month
AND created_at < '2013-10-25' -- AFter one month
AND pageview_url = '/cart';

-- CREATE TEMPORARY TABLE cart_session_w_nxt_pag_id
SELECT

```

```

cart_sessions.time_period,
cart_sessions.website_session_id,
cart_sessions.website_pageview_id,
MIN(website_pageviews.website_pageview_id) AS nxt_page
FROM cart_sessions
LEFT JOIN website_pageviews
ON cart_sessions.website_session_id = website_pageviews.website_session_id
AND website_pageviews.website_pageview_id > cart_sessions.website_pageview_id
GROUP BY 1,2
HAVING nxt_page IS NOT NULL;

CREATE TEMPORARY TABLE cart_sessions_with_orders
SELECT
time_period,
cart_sessions.website_session_id,
order_id,
items_purchased,
price_usd
FROM cart_sessions
INNER JOIN orders
ON cart_sessions.website_session_id = orders.website_session_id;

SELECT
time_period,
COUNT(DISTINCT item.website_session_id) AS Sessions,
SUM(clicked_to_another_page) AS clickthroughs,
SUM(clicked_to_another_page)/COUNT(DISTINCT item.website_session_id) AS cart_ctr,
SUM(items_purchased)/SUM(placed_orders) AS products_per_order,
SUM(price_usd)/SUM(placed_orders) as AOV,
SUM(price_usd)/COUNT(DISTINCT item.website_session_id) AS pev_per_cart_session
FROM
(
SELECT
cart_sessions.time_period,
cart_sessions.website_session_id,
CASE WHEN cart_sessions_w_nxt_pag_id.website_session_id IS NULL THEN 0 ELSE 1 END AS clicked_to_another_page,
CASE WHEN cart_sessions_with_orders.website_session_id IS NULL THEN 0 ELSE 1 END AS placed_orders,
items_purchased,
price_usd
FROM cart_sessions
LEFT JOIN cart_session_w_nxt_pag_id
ON cart_session_w_nxt_pag_id.website_session_id = cart_sessions.website_session_id
LEFT JOIN cart_sessions_with_orders
ON cart_sessions_with_orders.website_session_id = cart_sessions.website_session_id
) AS item
GROUP BY 1

```

	time_period	Sessions	clickthroughs	cart_ctr	products_per_order	AOV	pev_per_cart_session
►	A. Pre_Cross_sell	1830	1229	0.6716	1.0000	51.416380	18.318842
	B. Post_cross_sell	1975	1351	0.6841	1.0447	54.251848	18.431894

1 NEW MESSAGE
January 12, 2014

From: Cindy Sharp (CEO)
Subject: Recent Product Launch

Good morning,
On December 12th 2013, we launched a third product targeting the birthday gift market (Birthday Bear). Could you please run a pre-post analysis comparing the month before vs. the month after, in terms of session-to-order conversion rate, AOV, products per order, and revenue per session?
Thank you!
-Cindy

Result Preview

time_period	conv_rate	aov	products_per_order	revenue_per_session
A. Pre_Birthday_Bear				
B. Post_Birthday_Bear				

```

CREATE TEMPORARY TABLE sessions
SELECT
    website_session_id,
    CASE
        WHEN created_at < '2013-12-12' THEN 'A. Pre_Birthday_Bear'
        WHEN created_at >= '2013-12-12' THEN 'B. Post_Birthday_Bear'
        ELSE 'check date range'
    END AS time_period
FROM website_sessions
WHERE created_at BETWEEN '2013-11-12' AND '2014-01-12';

SELECT
    time_period,
    COUNT(DISTINCT sessions.website_session_id) As Sess,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT sessions.website_session_id) AS conv_rat,
    SUM(price_usd)/ COUNT(DISTINCT order_id) AS AOV,
    SUM(items_purchased) /COUNT(DISTINCT order_id) AS producst_per_order,
    SUM(price_usd) / COUNT(DISTINCT sessions.website_session_id) AS revenue_per_session
FROM sessions
LEFT JOIN orders
    ON sessions.website_session_id = orders.website_session_id
GROUP BY 1

```

	time_period	Sess	conv_rat	AOV	producst_per_order	revenue_per_session
▶	A. Pre_Birthday_Bear	17343	0.0608	54.226502	1.0464	3.298677
	B. Post_Birthday_Bear	13383	0.0702	56.931319	1.1234	3.998763

 NEW MESSAGE
October 15, 2014

From: Cindy Sharp (CEO)
Subject: Quality Issues & Refunds

Good morning,

Our Mr. Fuzzy supplier had some quality issues which weren't corrected until September 2013. Then they had a major problem where the bears' arms were falling off in Aug/Sep 2014. As a result, we replaced them with a new supplier on **September 16, 2014**.

Can you please pull **monthly product refund rates, by product, and confirm our quality issues are now fixed?**

-Cindy

[Reply](#) [Forward](#)

Result Preview

yr	mo	p1_orders	p1_refund_rt	p2_orders	p2_refund_rt	p3_orders	p3_refund_rt	p4_orders	p4_refund_rt
2012	3								
2012	4	60	0.0167	0	HULL	0	HULL	0	HULL
2012	5	99	0.0505	0	HULL	0	HULL	0	HULL
2012	6	108	0.0370	0	HULL	0	HULL	0	HULL
2012	7	140	0.0571	0	HULL	0	HULL	0	HULL
2012	8	169	0.0828	0	HULL	0	HULL	0	HULL
2012	9	228	0.0746	0	HULL	0	HULL	0	HULL
2012	10	287	0.0906	0	HULL	0	HULL	0	HULL
2012	11	371	0.0728	0	HULL	0	HULL	0	HULL
2012	12	618	0.0744	0	HULL	0	HULL	0	HULL
2013	1	506	0.0593	0	HULL	0	HULL	0	HULL
2013	2	343	0.0496	47	0.0213	0	HULL	0	HULL
2013	3	336	0.0714	162	0.0123	0	HULL	0	HULL
2013	4	320	0.0563	65	0.0462	0	HULL	0	HULL
2013	5	459	0.0414	94	0.0106	0	HULL	0	HULL
2013	6	489	0.0634	82	0.0244	0	HULL	0	HULL
2013	7	503	0.0775	90	0.0556	0	HULL	0	HULL
2013	8	509	0.0727	95	0.0316	0	HULL	0	HULL
2013	9								
2013	10								
2013	11								
2013	12								
2014	1								
2014	2								
2014	3								
2014	4								
2014	5								
2014	6								
2014	7								
2014	8								
2014	9								
2014	10								

```

SELECT
YEAR(order_items.created_at) AS Yr,
MONTH(order_items.created_at) AS Mo,
COUNT(DISTINCT CASE WHEN order_items.product_id = 1 THEN order_items.order_item_id ELSE NULL END) AS P1_orders,
COUNT(DISTINCT CASE WHEN order_items.product_id = 1 THEN order_item_refunds.order_item_refund_id ELSE NULL END)/
COUNT(DISTINCT CASE WHEN order_items.product_id = 1 THEN order_items.order_item_id ELSE NULL END) AS P1_refund_count,
COUNT(DISTINCT CASE WHEN order_items.product_id = 2 THEN order_items.order_item_id ELSE NULL END) AS P2_orders,
COUNT(DISTINCT CASE WHEN order_items.product_id = 2 THEN order_item_refunds.order_item_refund_id ELSE NULL END)/
COUNT(DISTINCT CASE WHEN order_items.product_id = 2 THEN order_items.order_item_id ELSE NULL END) AS P2_refund_count,
COUNT(DISTINCT CASE WHEN order_items.product_id = 3 THEN order_items.order_item_id ELSE NULL END) AS P3_orders,
COUNT(DISTINCT CASE WHEN order_items.product_id = 3 THEN order_item_refunds.order_item_refund_id ELSE NULL END)/
COUNT(DISTINCT CASE WHEN order_items.product_id = 3 THEN order_items.order_item_id ELSE NULL END) AS P3_refund_count,
COUNT(DISTINCT CASE WHEN order_items.product_id = 4 THEN order_items.order_item_id ELSE NULL END) AS P4_orders,
COUNT(DISTINCT CASE WHEN order_items.product_id = 4 THEN order_item_refunds.order_item_refund_id ELSE NULL END)/
COUNT(DISTINCT CASE WHEN order_items.product_id = 4 THEN order_items.order_item_id ELSE NULL END) AS P4_refund_count
FROM order_items
LEFT JOIN order_item_refunds
ON order_items.order_item_id = order_item_refunds.order_item_id
WHERE order_items.created_at <'2014-10-15'
GROUP BY 1,2;

```

	Yr	Mo	P1_orders	P1_refund_count	P2_orders	P2_refund_count	P3_orders	P3_refund_count	P4_orders	P4_refund_count	P1_refund_count
▶	2012	3	60	0.0167	0	HULL	0	HULL	0	HULL	HULL
	2012	4	99	0.0505	0	HULL	0	HULL	0	HULL	HULL
	2012	5	108	0.0370	0	HULL	0	HULL	0	HULL	HULL
	2012	6	140	0.0571	0	HULL	0	HULL	0	HULL	HULL
	2012	7	169	0.0828	0	HULL	0	HULL	0	HULL	HULL
	2012	8	228	0.0746	0	HULL	0	HULL	0	HULL	HULL
	2012	9	287	0.0906	0	HULL	0	HULL	0	HULL	HULL
	2012	10	371	0.0728	0	HULL	0	HULL	0	HULL	HULL
	2012	11	618	0.0744	0	HULL	0	HULL	0	HULL	HULL
	2012	12	506	0.0593	0	HULL	0	HULL	0	HULL	HULL
	2013	1	343	0.0496	47	0.0213	0	HULL	0	HULL	HULL
	2013	2	336	0.0714	162	0.0123	0	HULL	0	HULL	HULL
	2013	3	320	0.0563	65	0.0462	0	HULL	0	HULL	HULL
	2013	4	459	0.0414	94	0.0106	0	HULL	0	HULL	HULL
	2013	5	489	0.0634	82	0.0244	0	HULL	0	HULL	HULL
	2013	6	503	0.0775	90	0.0556	0	HULL	0	HULL	HULL
	2013	7	509	0.0727	95	0.0316	0	HULL	0	HULL	HULL

2013	8	510	0.0549	98	0.0102	0	NUL	0	NUL
2013	9	537	0.0428	98	0.0102	0	NUL	0	NUL
2013	10	603	0.0282	135	0.0148	0	NUL	0	NUL
2013	11	724	0.0345	174	0.0230	0	NUL	0	NUL
2013	12	818	0.0232	183	0.0219	139	0.0719	0	NUL
2014	1	728	0.0426	183	0.0219	200	0.0650	0	NUL
2014	2	584	0.0394	351	0.0171	211	0.0664	202	0.0099
2014	3	785	0.0306	193	0.0155	244	0.0697	205	0.0049
2014	4	917	0.0349	214	0.0187	267	0.0674	259	0.0154
2014	5	1030	0.0291	246	0.0163	299	0.0569	298	0.0067
2014	6	893	0.0571	245	0.0367	288	0.0556	249	0.0241
2014	7	961	0.0437	244	0.0369	276	0.0399	264	0.0152
2014	8	958	0.1378	237	0.0169	294	0.0680	303	0.0066
2014	9	1056	0.1326	251	0.0319	317	0.0662	327	0.0122
2014	10	513	0.0273	135	0.0074	165	0.0485	155	0.0323

1 NEW MESSAGE
November 01, 2014

From: Tom Parmesan (Marketing Director)
Subject: Repeat Visitors

Hey there,
We've been thinking about customer value based solely on their first session conversion and revenue. But if customers have repeat sessions, they may be more valuable than we thought. If that's the case, we might be able to spend a bit more to acquire them.
Could you please pull data on how many of our website visitors come back for another session? 2014 to date is good.
Thanks, Tom

Result Preview

Result Grid

repeat_sessions	users
0	
1	
2	
3	

```

CREATE TEMPORARY TABLE repeat_sessions_tab
SELECT
    new_session.user_id,
    new_session.website_session_id,
    website_sessions.website_session_id AS repeat_session
FROM
(
SELECT
    user_id,
    website_session_id
FROM website_sessions
WHERE created_at BETWEEN '2014-01-01' AND '2014-11-01'
AND is_repeat_session = 0
) AS new_session
LEFT JOIN website_sessions
ON website_sessions.user_id = new_session.user_id
    AND website_sessions.is_repeat_session = 1
    AND website_sessions.website_session_id > new_session.website_session_id
    AND website_sessions.created_at BETWEEN '2014-01-01' AND '2014-11-01'
;
SELECT *
FROM repeat_sessions_tab;
SELECT

```

```

repeat_sessions,
COUNT(DISTINCT user_id) AS users_count
FROM
(SELECT
user_id,
COUNT(DISTINCT website_session_id) as new_session,
COUNT(DISTINCT repeat_session) as repeat_sessions
FROM repeat_sessions_tab
GROUP BY 1
ORDER BY 3 DESC) AS new_repeat
GROUP BY 1
;

```

	repeat_sessions	users_count
0	126813	
1	14086	
2	315	
3	4686	

@October 9, 2022

Result Preview

avg_days_first_to_second	min_days_first_to_second	max_days_first_to_second
[Redacted]	[Redacted]	[Redacted]

```

CREATE TEMPORARY TABLE new_session_w_repeat_session
SELECT
first_sessions.user_id,
first_sessions.website_session_id AS new_session,
first_sessions.created_at AS first_created_at,
MIN(website_sessions.website_session_id) AS repeat_session,
MIN(website_sessions.created_at) AS repeat_created_at
FROM
(
SELECT
website_session_id,
created_at,
user_id
FROM website_sessions
WHERE created_at BETWEEN '2014-01-01' AND '2014-11-03'
AND is_repeat_session = 0
) AS first_sessions
LEFT JOIN website_sessions
ON first_sessions.user_id = website_sessions.user_id
AND is_repeat_session = 1
AND website_sessions.website_session_id > first_sessions.website_session_id
GROUP BY 1
HAVING repeat_session IS NOT NULL;

```

```

SELECT
    AVG(days_gap) AS avg_days_gap,
    MIN(days_gap) AS Min_days_gap,
    MAX(days_gap) AS Max_days_gap
FROM
(
SELECT
    user_id,
    DATEDIFF(repeat_created_at, first_created_at) AS days_gap
FROM new_session_w_repeat_session) AS days
;

```

	avg_days_gap	Min_days_gap	Max_days_gap
▶	34.7002	1	69

NEW MESSAGE
November 05, 2014

From: Tom Parmesan (Marketing Director)
Subject: Repeat Channel Mix

Hi there,
Let's do a bit more digging into our repeat customers.
Can you help me understand the channels they come back through? Curious if it's all direct type-in, or if we're paying for these customers with paid search ads multiple times.
Comparing new vs. repeat sessions by channel would be really valuable, if you're able to pull it! 2014 to date is great.
Thanks, Tom

Reply Forward

Result Preview

channel_group	new_sessions	repeat_sessions
▶ organic_search		
paid_brand		
direct_type_in		
paid_nonbrand		
paid_social		

```

SELECT
CASE
    WHEN http_referer IN ('https://www.gsearch.com', 'https://www.bsearch.com') AND utm_source IS NULL THEN 'Organic_search'
    WHEN utm_campaign = 'brand' THEN 'Paid_brand'
    WHEN utm_campaign IS NULL AND http_referer IS NULL THEN 'Direct_type_in'
    WHEN utm_campaign = 'nonbrand' THEN 'Paid_nonbrand'
    WHEN utm_source = 'socialbook' THEN 'Paid_social'
    ELSE 'check_logic'
END AS channel_group,
COUNT(DISTINCT CASE WHEN is_repeat_session = 0 THEN website_session_id ELSE NULL END) AS new_sessions,
COUNT(DISTINCT CASE WHEN is_repeat_session = 1 THEN website_session_id ELSE NULL END) AS repeat_sessions
FROM website_sessions
WHERE created_at BETWEEN '2014-01-01' AND '2014-11-05'
GROUP BY 1
ORDER BY 3 DESC

```

	channel_group	new_sessions	repeat_sessions
▶	Organic_search	7139	11507
	Paid_brand	6432	11027
	Direct_type_in	6591	10564
	Paid_nonbrand	119950	0
	Paid_social	7652	0

NEW MESSAGE
November 08, 2014

From: **Morgan Rockwell (Website Manager)**
Subject: **Top Website Pages**

Hi there!

Sounds like you and Tom have learned a lot about our repeat customers. Can I trouble you for one more thing?

I'd love to do a **comparison of conversion rates and revenue per session for repeat sessions vs new sessions**.

Let's continue using data from **2014, year to date**.

Thank you!
-Morgan

[Reply](#) [Forward](#)

Result Preview

Result Grid Filter Rows: Search

is_repeat_session	sessions	conv_rate	rev_per_session
0			
1			

```
SELECT
    is_repeat_session,
    COUNT(DISTINCT website_sessions.website_session_id) as sessions,
    COUNT(DISTINCT order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conv_rt,
    SUM(price_usd)/COUNT(DISTINCT website_sessions.website_session_id) AS rev_per_session
FROM website_sessions
LEFT JOIN orders
    ON website_sessions.website_session_id = orders.website_session_id
WHERE website_sessions.created_at BETWEEN '2014-01-01' AND '2014-11-08'
GROUP BY 1;
```

	is_repeat_session	sessions	conv_rt	rev_per_session
▶	0	149787	0.0680	4.343754
	1	33577	0.0811	5.168828

Final Project