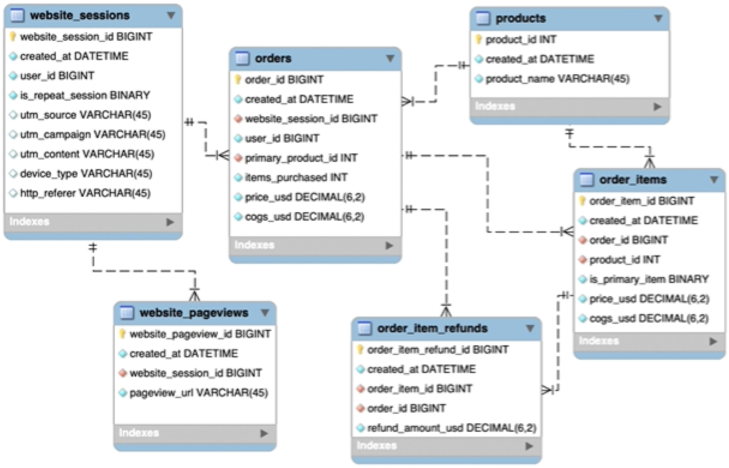
Advance MySQL Udemy Course

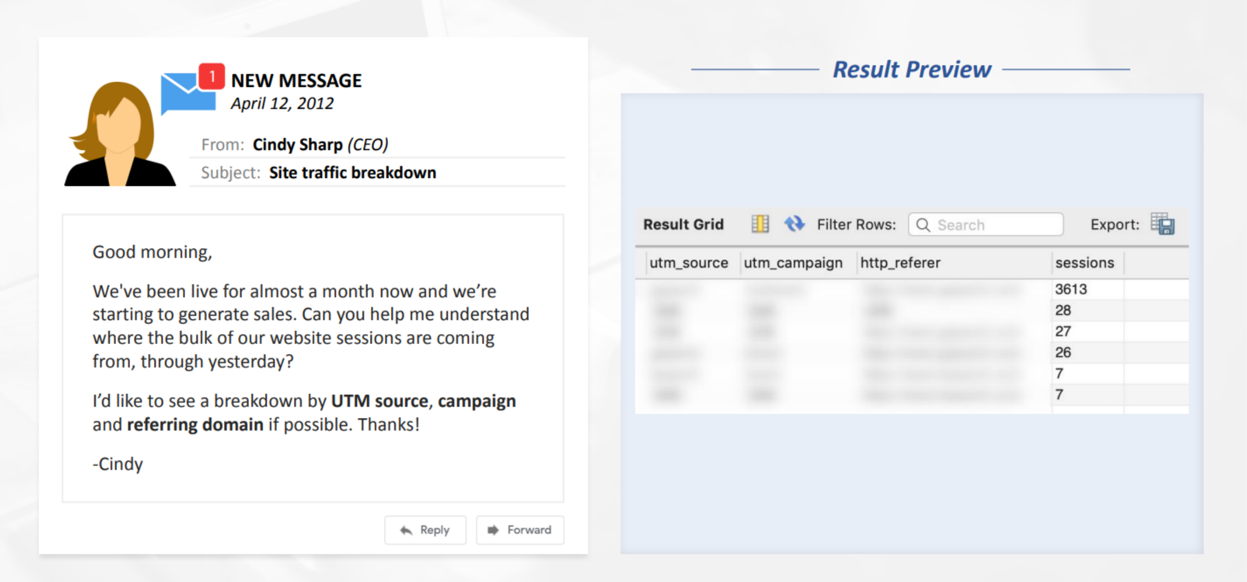


We will be working with six related tables, which contain eCommerce data about:

- **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?

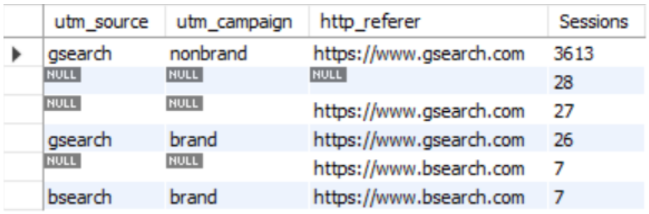
I'd like to see a breakdown by **UTM source**, **campaign** and **referring domain** if possible. Thanks!

-Cindy

utm_source	utm_campaign	http_referer	sessions
[illegible]	[illegible]	[illegible]	3613
[illegible]	[illegible]	[illegible]	28
[illegible]	[illegible]	[illegible]	27
[illegible]	[illegible]	[illegible]	26
[illegible]	[illegible]	[illegible]	7
[illegible]	[illegible]	[illegible]	7

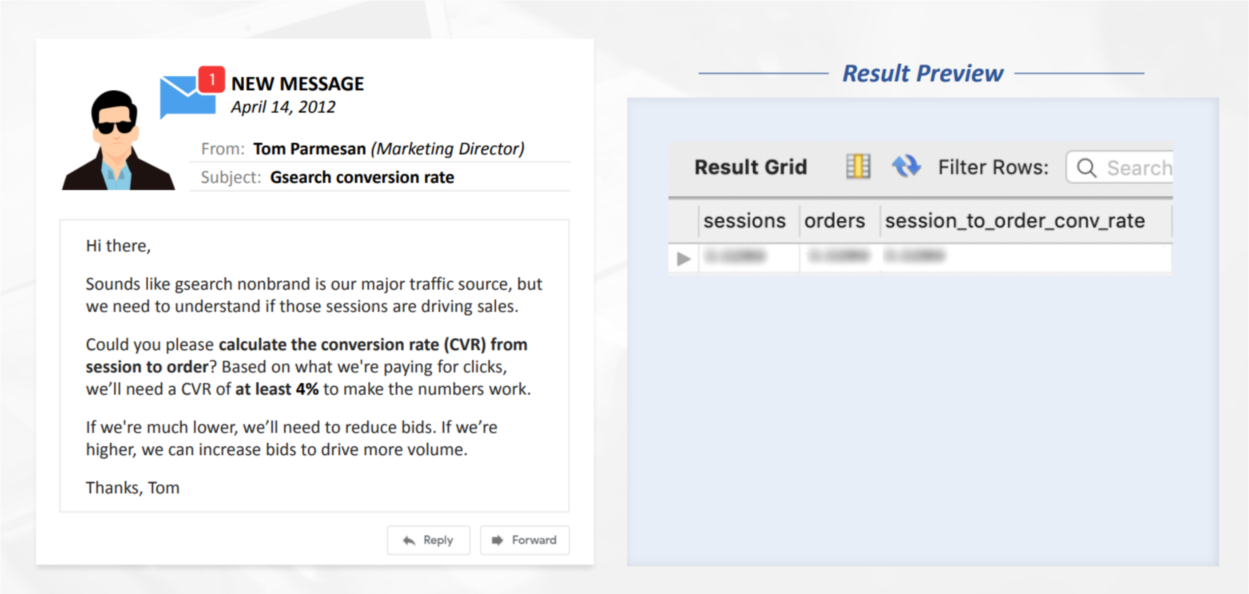
```
SELECT
    utm_source,
    utm_campaign,
    http_referer,
    COUNT(DISTINCT website_session_id) as Sessions
FROM website_sessions
WHERE created_at < '2012-04-12'
GROUP BY
  utm_source,
    utm_campaign,
    http_referer
ORDER BY 4 DESC
```

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.



NEW MESSAGE
April 14, 2012

From: **Tom Parmesan** *(Marketing Director)*
Subject: **Gsearch conversion rate**

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.

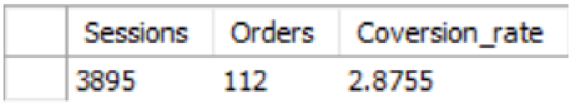
If we're much lower, we'll need to reduce bids. If we're higher, we can increase bids to drive more volume.

Thanks, Tom

sessions	orders	session_to_order_conv_rate
[illegible]	[illegible]	[illegible]

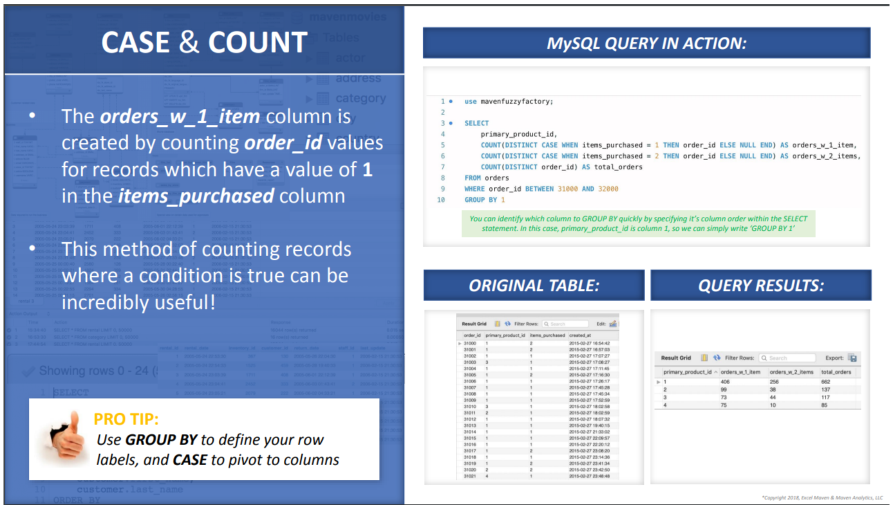
```
SELECT
  COUNT(distinct website_sessions.website_session_id) as Sessions,
  COUNT(DISTINCT orders.order_id) as Orders,
  (COUNT(DISTINCT orders.order_id)/COUNT(distinct website_sessions.website_session_id))*100 as Coversion_rate
FROM website_sessions
 LEFT JOIN orders
 ON orders.website_session_id = 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	Coversion_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!

PRO TIP:

*Use **GROUP BY** to define your row labels, and **CASE** to pivot to columns*

MySQL QUERY IN ACTION:

```
use mavenfuzzyfactory;

SELECT
    primary_product_id,
    COUNT(DISTINCT CASE WHEN items_purchased = 1 THEN order_id ELSE NULL END) AS orders_w_1_item,
    COUNT(DISTINCT CASE WHEN items_purchased = 2 THEN order_id ELSE NULL END) AS orders_w_2_items,
    COUNT(DISTINCT order_id) AS total_orders
FROM orders
WHERE order_id BETWEEN 31000 AND 32000
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'

ORIGINAL TABLE:

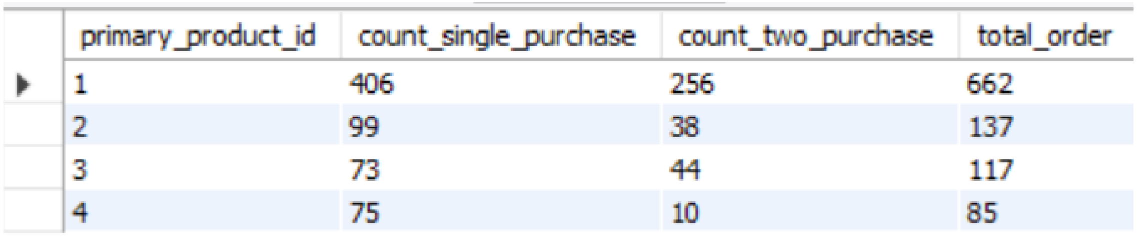
order_id	primary_product_id	items_purchased	created_at
31000	1	2	2015-02-27 16:54:42
31001	1	2	2015-02-27 16:57:03
31002	1	1	2015-02-27 17:07:27
31003	3	1	2015-02-27 17:08:27
31004	1	1	2015-02-27 17:11:45
31005	1	2	2015-02-27 17:16:30
31006	1	1	2015-02-27 17:26:17
31007	1	1	2015-02-27 17:45:28
31008	1	1	2015-02-27 17:45:34
31009	1	1	2015-02-27 17:52:59
31010	3	1	2015-02-27 18:02:58
31011	2	1	2015-02-27 18:02:59
31012	1	1	2015-02-27 18:07:32
31013	1	1	2015-02-27 19:40:15
31014	1	1	2015-02-27 21:33:02
31015	1	1	2015-02-27 22:09:57
31016	1	1	2015-02-27 22:20:12
31017	1	2	2015-02-27 23:08:20
31018	1	1	2015-02-27 23:14:36
31019	1	2	2015-02-27 23:41:34
31020	2	2	2015-02-27 23:42:50
31021	4	1	2015-02-27 23:48:48

QUERY RESULTS:

primary_product_id	orders_w_1_item	orders_w_2_items	total_orders
1	406	256	662
2	99	38	137
3	73	44	117
4	75	10	85

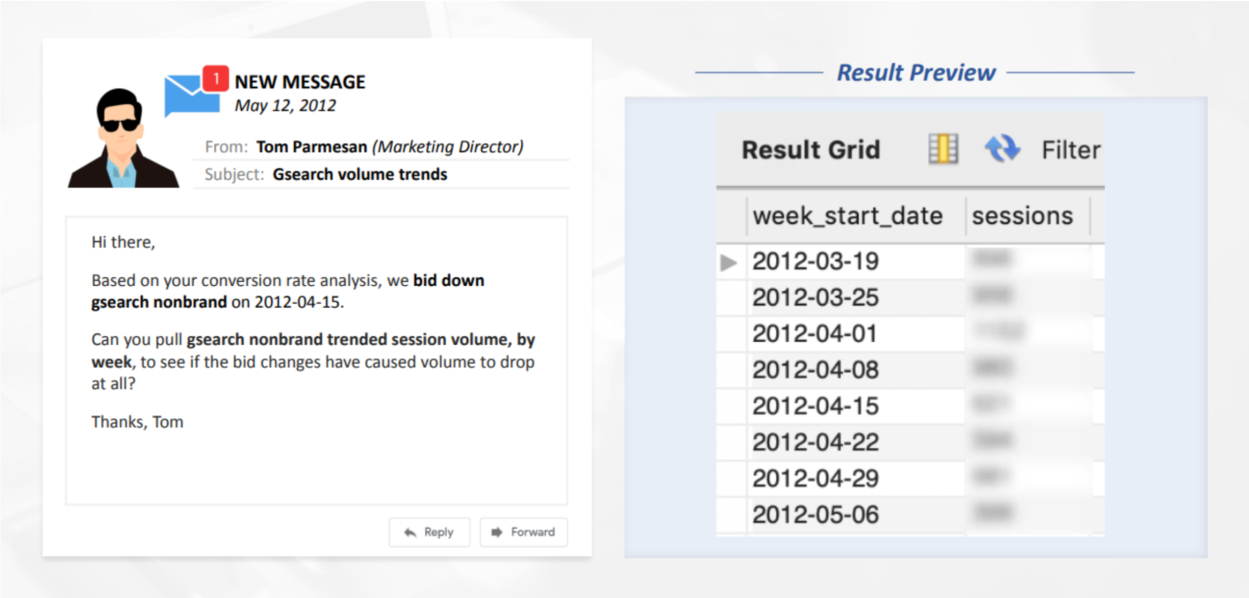
```
SELECT
    primary_product_id,
    count(CASE WHEN items_purchased = 1 THEN order_id ELSE NULL END) AS count_single_purchase,
    count(CASE WHEN items_purchased = 2 THEN order_id ELSE NULL END) AS count_two_purchase,
    count(distinct order_id) as total_order
FROM mavenfuzzyfactory.orders
WHERE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



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

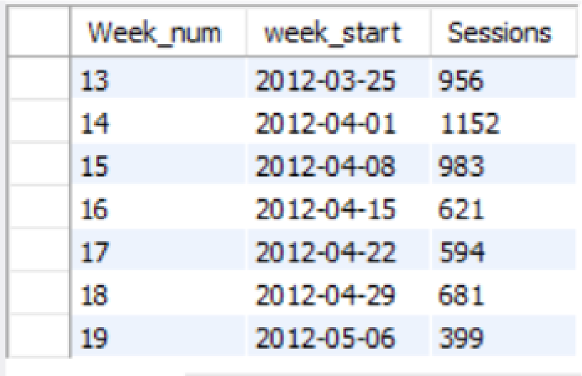
Result Preview

Result Grid Filter

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

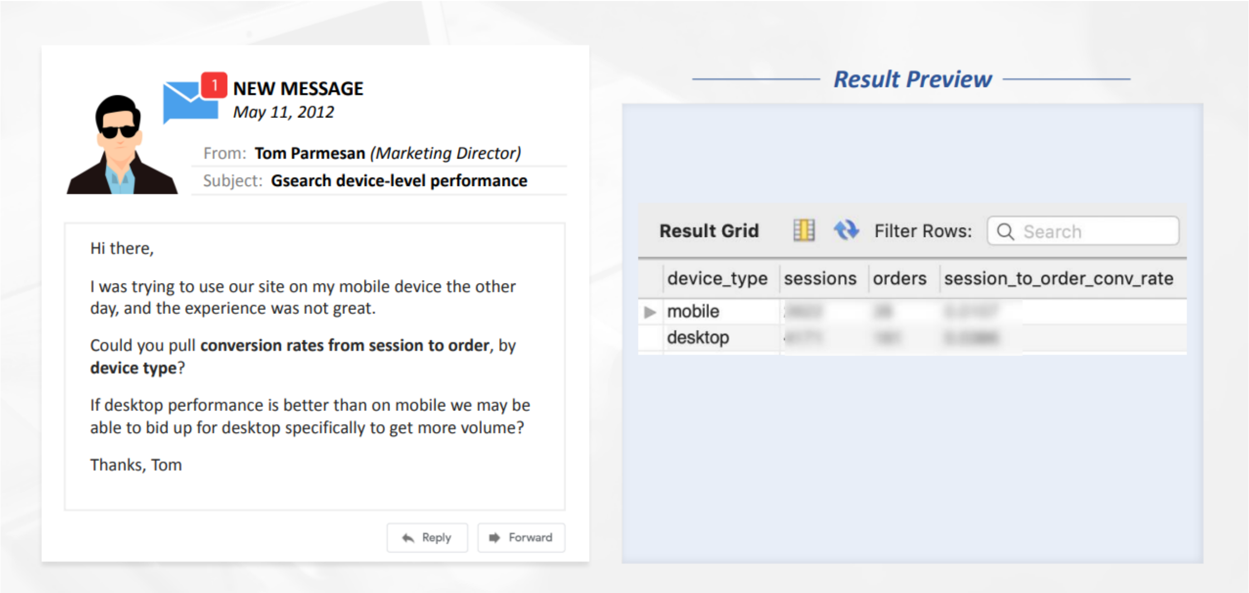
```
SELECT
  week(created_at) as Week_num,
  MIN(date(created_at)) as week_start,
  count(DISTINCT  website_session_id) as Sessions
FROM website_sessions
WHERE created_at < '2012-05-10'
AND utm_campaign = 'nonbrand'
AND utm_source = 'gsearch'
GROUP BY 1;
```

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



NEW MESSAGE
May 11, 2012

From: **Tom Parmesan** *(Marketing Director)*
Subject: **Gsearch device-level performance**

Hi there,

I was trying to use our site on my mobile device the other day, and the experience was not great.

Could you pull **conversion rates from session to order**, by **device type**?

If desktop performance is better than on mobile we may be able to bid up for desktop specifically to get more volume?

Thanks, Tom

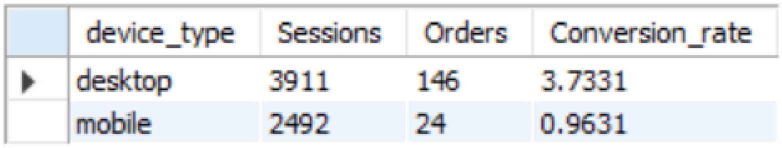
Result Preview

Result Grid Filter Rows: Search

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

```
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