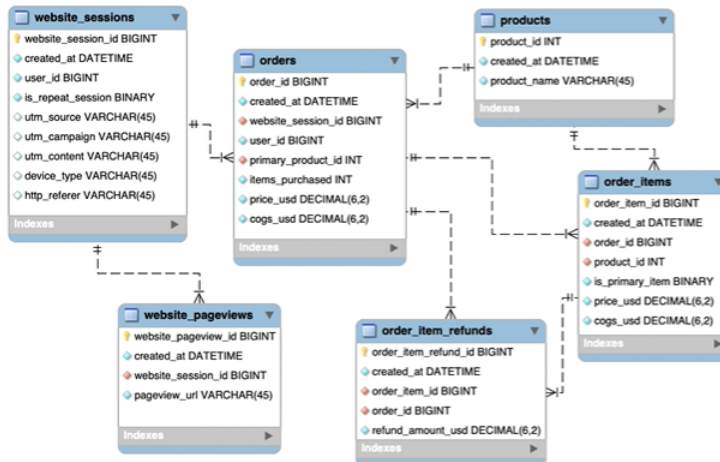




# Final Project



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

First, I'd like to show our volume growth. Can you pull overall session and order volume, trended by quarter for the life of the business? Since the most recent quarter is incomplete, you can decide how to handle it.

```
SELECT
  YEAR(website_sessions.created_at) AS Yr,
  QUARTER(website_sessions.created_at) AS Qtr,
  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
GROUP BY 1,2;
```

	Yr	Qtr	Sessions	Orders
▶	2012	1	1879	60
	2012	2	11433	347
	2012	3	16892	684
	2012	4	32266	1495
	2013	1	19833	1273
	2013	2	24745	1718
	2013	3	27663	1840
	2013	4	40540	2616
	2014	1	46779	3069
	2014	2	53129	3848
	2014	3	57141	4035
	2014	4	76373	5908
	2015	1	64198	5420

2

Next, let's showcase all of our efficiency improvements. I would love to show quarterly figures since we launched, for session-to-order conversion rate, revenue per order, and revenue per session.

```

SELECT
  YEAR(website_sessions.created_at) AS Yr,
  QUARTER(website_sessions.created_at) AS Qtr,
  COUNT(DISTINCT order_id)/COUNT(DISTINCT website_sessions.website_session_id) AS conv_rt,
  ROUND(SUM(price_usd)/COUNT(DISTINCT order_id),2) AS Revnue_per_order,
  ROUND(SUM(price_usd)/COUNT(DISTINCT website_sessions.website_session_id),2) AS Revenue_per_session
FROM website_sessions
LEFT JOIN orders
  ON website_sessions.website_session_id = orders.website_session_id
GROUP BY 1,2;

```

	Yr	Qtr	conv_rt	Revnue_per_order	Revenue_per_session
►	2012	1	0.0319	49.99	1.60
	2012	2	0.0304	49.99	1.52
	2012	3	0.0405	49.99	2.02
	2012	4	0.0463	49.99	2.32
	2013	1	0.0642	52.14	3.35
	2013	2	0.0694	51.54	3.58
	2013	3	0.0665	51.73	3.44
	2013	4	0.0645	54.72	3.53
	2014	1	0.0656	62.16	4.08
	2014	2	0.0724	64.37	4.66
	2014	3	0.0706	64.49	4.55
	2014	4	0.0774	63.79	4.93
	2015	1	0.0844	62.80	5.30

3

I'd like to show how we've grown specific channels. Could you pull a quarterly view of orders from Gsearch nonbrand, Bsearch nonbrand, brand search overall, organic search, and direct type-in?

```

SELECT
  YEAR(website_sessions.created_at) AS Yr,
  QUARTER(website_sessions.created_at) AS Qtr,
  CASE
    WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN 'Gsearch_nonbrand'
    WHEN utm_source = 'Bsearch' AND utm_campaign = 'nonbrand' THEN 'Bsearch_nonbrand'
    WHEN utm_campaign = 'brand' THEN 'Brand_search_overall'
    WHEN utm_source IS NULL AND http_referer IN ('https://www.gsearch.com', 'https://www.bsearch.com') THEN 'Organic'
    WHEN http_referer IS NULL THEN 'Direct_type_in'
    ELSE 'check_logic'
  END AS 'channel_type',
  COUNT(DISTINCT order_id) AS Orders
FROM website_sessions
LEFT JOIN orders
  ON website_sessions.website_session_id = orders.website_session_id
GROUP BY 1,2,3;

```

	Yr	Qtr	channel_type	Orders
	2012	1	Brand_search_overall	0
	2012	1	Direct_type_in	0
	2012	1	Gsearch_nonbrand	60
►	2012	1	Organic	0
	2012	2	Brand_search_overall	20
	2012	2	Direct_type_in	21
	2012	2	Gsearch_nonbrand	291
	2012	2	Organic	15
	2012	3	Bsearch_nonbrand	82
	2012	3	Brand_search_overall	48
	2012	3	Direct_type_in	32
	2012	3	Gsearch_nonbrand	482
	2012	3	Organic	40
	2012	4	Bsearch_nonbrand	311
	2012	4	Brand_search_overall	88
	2012	4	Direct_type_in	89
	2012	4	Gsearch_nonbrand	913
	2012	4	Organic	94

	Yr	Qtr	channel_type	Orders
	2013	1	Bsearch_nonbrand	183
	2013	1	Brand_search_overall	108
	2013	1	Direct_type_in	91
	2013	1	Gsearch_nonbrand	766
	2013	1	Organic	125
	2013	2	Bsearch_nonbrand	237
	2013	2	Brand_search_overall	114
	2013	2	Direct_type_in	119
	2013	2	Gsearch_nonbrand	1114
	2013	2	Organic	134
	2013	3	Bsearch_nonbrand	245
	2013	3	Brand_search_overall	153
	2013	3	Direct_type_in	143
	2013	3	Gsearch_nonbrand	1132
	2013	3	Organic	167
	2013	4	Bsearch_nonbrand	291
	2013	4	Brand_search_overall	248
	2013	4	Direct_type_in	197
	2013	4	Gsearch nonbrand	1657

	Yr	Qtr	channel_type	Orders
	2013	4	Organic	223
	2014	1	Bsearch_nonbrand	344
	2014	1	Brand_search_overall	354
	2014	1	check_logic	55
	2014	1	Direct_type_in	311
	2014	1	Gsearch_nonbrand	1667
	2014	1	Organic	338
	2014	2	Bsearch_nonbrand	427
	2014	2	Brand_search_overall	410
	2014	2	Direct_type_in	367
	2014	2	Gsearch_nonbrand	2208
	2014	2	Organic	436
	2014	3	Bsearch_nonbrand	434
	2014	3	Brand_search_overall	432
	2014	3	check_logic	63
	2014	3	Direct_type_in	402
	2014	3	Gsearch_nonbrand	2259
	2014	3	Organic	445
	2014	4	Bsearch nonbrand	683

	2014	4	Brand_search_overall	615
	2014	4	check_logic	225
	2014	4	Direct_type_in	532
	2014	4	Gsearch_nonbrand	3248
	2014	4	Organic	605
	2015	1	Bsearch_nonbrand	581
	2015	1	Brand_search_overall	622
	2015	1	Direct_type_in	552
	2015	1	Gsearch_nonbrand	3025
	2015	1	Organic	640

```

SELECT
  YEAR(sess.created_at) AS Yr,
  QUARTER(sess.created_at) AS Qtr,
  COUNT(DISTINCT CASE WHEN channel_type='Gsearch_nonbrand' THEN order_id ELSE NULL END) AS Gsearch_nonbarnd_orders,
  COUNT(DISTINCT CASE WHEN channel_type='Bsearch_nonbrand' THEN order_id ELSE NULL END) AS Bsearch_nonbarnd_orders,
  COUNT(DISTINCT CASE WHEN channel_type='Brand_search_overall' THEN order_id ELSE NULL END) AS Brand_overall_orders,
  COUNT(DISTINCT CASE WHEN channel_type='Organic' THEN order_id ELSE NULL END) AS Organic_orders,
  COUNT(DISTINCT CASE WHEN channel_type='Direct_type_in' THEN order_id ELSE NULL END) AS Direct_type_in_orders
FROM
(
  SELECT
    website_session_id,
    created_at,
    CASE
      WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN 'Gsearch_nonbrand'
      WHEN utm_source = 'Bsearch' AND utm_campaign = 'nonbrand' THEN 'Bsearch_nonbrand'
      WHEN utm_campaign = 'brand' THEN 'Brand_search_overall'
      WHEN utm_source IS NULL AND http_referer IN ('https://www.gsearch.com', 'https://www.bsearch.com') THEN 'Organic'
      WHEN http_referer IS NULL THEN 'Direct_type_in'
      ELSE 'check_logic'
    END AS 'channel_type'
  FROM website_sessions
) AS sess
LEFT JOIN orders
  ON sess.website_session_id = orders.website_session_id
GROUP BY 1,2
;

```

	Yr	Qtr	Gsearch_nonbarnd_orders	Bsearch_nonbarnd_orders	Brand_overall_orders	Organic_orders	Direct_type_in_orders
▶	2012	1	60	0	0	0	0
	2012	2	291	0	20	15	21
	2012	3	482	82	48	40	32
	2012	4	913	311	88	94	89
	2013	1	766	183	108	125	91
	2013	2	1114	237	114	134	119
	2013	3	1132	245	153	167	143
	2013	4	1657	291	248	223	197
	2014	1	1667	344	354	338	311
	2014	2	2208	427	410	436	367
	2014	3	2259	434	432	445	402
	2014	4	3248	683	615	605	532
	2015	1	3025	581	622	640	552

4

Next, let's show the overall session-to-order conversion rate trends for those same channels, by quarter. Please also make a note of any periods where we made major improvements or optimizations.

```

SELECT
  YEAR(sess.created_at) AS Yr,
  QUARTER(sess.created_at) AS Qtr,
  COUNT(DISTINCT CASE WHEN channel_type='Gsearch_nonbrand' THEN order_id ELSE NULL END)
  / COUNT(DISTINCT CASE WHEN channel_type='Gsearch_nonbrand' THEN sess.website_session_id ELSE NULL END) AS Gsearch_nonbarnd_conv_rt,
  COUNT(DISTINCT CASE WHEN channel_type='Bsearch_nonbrand' THEN order_id ELSE NULL END)
  / COUNT(DISTINCT CASE WHEN channel_type='Bsearch_nonbrand' THEN sess.website_session_id ELSE NULL END) AS Bsearch_nonbarnd_conv_rt,
  COUNT(DISTINCT CASE WHEN channel_type='Brand_search_overall' THEN order_id ELSE NULL END)
  / COUNT(DISTINCT CASE WHEN channel_type='Brand_search_overall' THEN sess.website_session_id ELSE NULL END) AS Brand_overall_conv_rt,
  COUNT(DISTINCT CASE WHEN channel_type='Organic' THEN order_id ELSE NULL END)
  / COUNT(DISTINCT CASE WHEN channel_type='Organic' THEN sess.website_session_id ELSE NULL END) AS Organic_conv_rt,
  COUNT(DISTINCT CASE WHEN channel_type='Direct_type_in' THEN order_id ELSE NULL END)
  / COUNT(DISTINCT CASE WHEN channel_type='Direct_type_in' THEN sess.website_session_id ELSE NULL END) AS Direct_type_in_conv_rt
FROM
(
  SELECT
    website_session_id,
    created_at,
    CASE

```

```

    WHEN utm_source = 'gsearch' AND utm_campaign = 'nonbrand' THEN 'Gsearch_nonbrand'
    WHEN utm_source = 'Bsearch' AND utm_campaign = 'nonbrand' THEN 'Bsearch_nonbrand'
    WHEN utm_campaign = 'brand' THEN 'Brand_search_overall'
    WHEN utm_source IS NULL AND http_referer IN ('https://www.gsearch.com', 'https://www.bsearch.com') THEN 'Organic'
    WHEN http_referer IS NULL THEN 'Direct_type_in'
    ELSE 'check_logic'
  END AS 'channel_type'
FROM website_sessions
) AS sess
LEFT JOIN orders
  ON sess.website_session_id = orders.website_session_id
GROUP BY 1,2
;

```

	Yr	Qtr	Gsearch_nonbarnd_conv_rt	Bsearch_nonbarnd_conv_rt	Brand_overall_conv_rt	Organic_conv_rt	Direct_type_in_conv_rt
►	2012	1	0.0324	NULL	0.0000	0.0000	0.0000
	2012	2	0.0284	NULL	0.0526	0.0359	0.0536
	2012	3	0.0384	0.0408	0.0602	0.0498	0.0443
	2012	4	0.0436	0.0497	0.0531	0.0539	0.0537
	2013	1	0.0612	0.0693	0.0703	0.0753	0.0614
	2013	2	0.0685	0.0690	0.0679	0.0760	0.0735
	2013	3	0.0639	0.0697	0.0703	0.0734	0.0719
	2013	4	0.0629	0.0601	0.0801	0.0694	0.0647
	2014	1	0.0693	0.0704	0.0839	0.0756	0.0765
	2014	2	0.0702	0.0695	0.0804	0.0797	0.0738
	2014	3	0.0703	0.0698	0.0756	0.0733	0.0702
	2014	4	0.0782	0.0841	0.0812	0.0784	0.0748
	2015	1	0.0861	0.0850	0.0852	0.0821	0.0775

	Qtr	Gsearch_nonbarnd_conv_rt	Bsearch_nonbarnd_conv_rt	Brand_overall_conv_rt	Organic_conv_rt	Direct_type_in_conv_rt
►	1	0.0750	0.0771	0.0830	0.0792	0.0752
	2	0.0624	0.0693	0.0760	0.0765	0.0726
	3	0.0621	0.0648	0.0729	0.0712	0.0684
	4	0.0655	0.0668	0.0771	0.0727	0.0692

**5** We've come a long way since the days of selling a single product. Let's pull monthly trending for revenue and margin by product, along with total sales and revenue. Note anything you notice about seasonality.

```

SELECT
  YEAR(created_at) AS Yr,
  MONTH(created_at) AS Mo,
  SUM(CASE WHEN product_id = 1 THEN price_usd ELSE NULL END) AS Mr_fuzzy,
  SUM(CASE WHEN product_id = 2 THEN price_usd ELSE NULL END) AS Love_Bear,
  SUM(CASE WHEN product_id = 3 THEN price_usd ELSE NULL END) AS Sugar_Panda,
  SUM(CASE WHEN product_id = 4 THEN price_usd ELSE NULL END) AS Mini_Bear,
  SUM(price_usd) as Revenue,
  SUM(price_usd - cogs_usd) AS Margin
FROM order_items
GROUP BY 1,2;

```

	Yr	Mo	Mr_fuzzy	Love_Bear	Sugar_Panda	Mini_Bear	Revenue	Margin
▶	2012	3	2999.40	NULL	NULL	NULL	2999.40	1830.00
	2012	4	4949.01	NULL	NULL	NULL	4949.01	3019.50
	2012	5	5398.92	NULL	NULL	NULL	5398.92	3294.00
	2012	6	6998.60	NULL	NULL	NULL	6998.60	4270.00
	2012	7	8448.31	NULL	NULL	NULL	8448.31	5154.50
	2012	8	11397.72	NULL	NULL	NULL	11397.72	6954.00
	2012	9	14347.13	NULL	NULL	NULL	14347.13	8753.50
	2012	10	18546.29	NULL	NULL	NULL	18546.29	11315.50
	2012	11	30893.82	NULL	NULL	NULL	30893.82	18849.00
	2012	12	25294.94	NULL	NULL	NULL	25294.94	15433.00
	2013	1	17146.57	2819.53	NULL	NULL	19966.10	12224.00
	2013	2	16796.64	9718.38	NULL	NULL	26515.02	16323.00
	2013	3	15996.80	3899.35	NULL	NULL	19896.15	12197.50
	2013	4	22945.41	5639.06	NULL	NULL	28584.47	17524.50
	2013	5	24445.11	4919.18	NULL	NULL	29364.29	17989.50
	2013	6	25144.97	5399.10	NULL	NULL	30544.07	18716.50

	Yr	Mo	Mr_fuzzy	Love_Bear	Sugar_Panda	Mini_Bear	Revenue	Margin
	2013	7	25444.91	5699.05	NULL	NULL	31143.96	19087.00
	2013	8	25494.90	5879.02	NULL	NULL	31373.92	19230.00
	2013	9	26844.63	5879.02	NULL	NULL	32723.65	20053.50
	2013	10	30143.97	8098.65	NULL	NULL	38242.62	23454.00
	2013	11	36192.76	10438.26	NULL	NULL	46631.02	28607.00
	2013	12	40891.82	10978.17	6392.61	NULL	58262.60	36190.00
	2014	1	36392.72	10978.17	9198.00	NULL	56568.89	35366.50
	2014	2	29194.16	21056.49	9703.89	6057.98	66012.52	41762.00
	2014	3	39242.15	11578.07	11221.56	6147.95	68189.73	43068.50
	2014	4	45840.83	12837.86	12279.33	7767.41	78725.43	49713.50
	2014	5	51489.70	14757.54	13751.01	8937.02	88935.27	56167.50
	2014	6	44641.07	14697.55	13245.12	7467.51	80051.25	50600.50
	2014	7	48040.39	14637.56	12693.24	7917.36	83288.55	52566.50
	2014	8	47890.42	14217.63	13521.06	9086.97	84716.08	53579.00
	2014	9	52789.44	15057.49	14578.83	9806.73	92232.49	58309.50
	2014	10	58638.27	17037.16	16924.32	11306.23	103905.98	65747.00

	Yr	Mo	Mr_fuzzy	Love_Bear	Sugar_Panda	Mini_Bear	Revenue	Margin
	2014	11	72535.49	22616.23	19545.75	13465.51	128162.98	80985.00
	2014	12	79184.16	23216.13	24788.61	17634.12	144823.02	91857.00
	2015	1	69586.08	23636.06	20695.50	18293.90	132211.54	83911.00
	2015	2	55638.87	38633.56	18625.95	16314.56	129212.94	82006.00
	2015	3	43191.36	13377.77	12095.37	10286.57	78951.07	50030.50

**6** Let's dive deeper into the impact of introducing new products. Please pull monthly sessions to the /products page, and show how the % of those sessions clicking through another page has changed over time, along with a view of how conversion from /products to placing an order has improved.

```
CREATE TEMPORARY TABLE products_sessions
SELECT
  website_session_id,
  website_pageview_id,
  created_at as first_created_at
FROM website_pageviews
WHERE pageview_url = '/products';

SELECT
  YEAR(products_sessions.first_created_at) AS Yr,
  MONTH(products_sessions.first_created_at) AS Mo,
  COUNT(DISTINCT products_sessions.website_session_id) AS product_sessions,
  COUNT(DISTINCT website_pageviews.website_session_id) AS next_sessions,
  COUNT(DISTINCT website_pageviews.website_session_id)/COUNT(DISTINCT products_sessions.website_session_id) AS product_clickthrough_rt,
  COUNT(DISTINCT order_id) AS Orders,
  COUNT(DISTINCT order_id)/COUNT(DISTINCT products_sessions.website_session_id) product_pg_to_order_conv_rt
FROM products_sessions
LEFT JOIN website_pageviews
  ON website_pageviews.website_session_id = products_sessions.website_session_id
  AND website_pageviews.website_pageview_id > products_sessions.website_pageview_id
LEFT JOIN orders
```

ON products\_sessions.website\_session\_id = orders.website\_session\_id

GROUP BY 1,2;

	Yr	Mo	product_sessions	next_sessions	product_clickthrough_rt	Orders	product_pg_to_order_conv_rt
►	2012	3	743	530	0.7133	60	0.0808
	2012	4	1447	1029	0.7111	99	0.0684
	2012	5	1584	1135	0.7165	108	0.0682
	2012	6	1752	1247	0.7118	140	0.0799
	2012	7	2018	1438	0.7126	169	0.0837
	2012	8	3012	2198	0.7297	228	0.0757
	2012	9	3126	2258	0.7223	287	0.0918
	2012	10	4030	2948	0.7315	371	0.0921
	2012	11	6743	4849	0.7191	618	0.0917
	2012	12	5013	3620	0.7221	506	0.1009
	2013	1	3380	2595	0.7678	391	0.1157
	2013	2	3685	2803	0.7607	497	0.1349
	2013	3	3371	2576	0.7642	385	0.1142
	2013	4	4362	3356	0.7694	553	0.1268
	2013	5	4684	3609	0.7705	571	0.1219
	2013	6	4600	3536	0.7687	594	0.1291
	2013	7	5020	3890	0.7749	603	0.1201
	2013	8	5226	3951	0.7560	608	0.1163
	2013	9	5399	4072	0.7542	629	0.1165
	2013	10	6038	4564	0.7559	708	0.1173
	2013	11	7886	5900	0.7482	861	0.1092
	2013	12	8840	7026	0.7948	1047	0.1184
	2014	1	7790	6387	0.8199	983	0.1262
	2014	2	7960	6485	0.8147	1021	0.1283
	2014	3	8110	6669	0.8223	1065	0.1313
	2014	4	9744	7958	0.8167	1241	0.1274
	2014	5	10261	8465	0.8250	1368	0.1333
	2014	6	10011	8260	0.8251	1239	0.1238
	2014	7	10837	8958	0.8266	1287	0.1188
	2014	8	10768	8980	0.8340	1324	0.1230
	2014	9	11128	9156	0.8228	1424	0.1280
	2014	10	12335	10235	0.8298	1609	0.1304
	2014	11	14476	12020	0.8303	1985	0.1371
	2014	12	17240	14609	0.8474	2314	0.1342
	2015	1	15217	12992	0.8538	2099	0.1379
	2015	2	14373	12187	0.8479	2067	0.1438
	2015	3	9022	7723	0.8560	1254	0.1390

7  
~ 18:40

We made our 4<sup>th</sup> product available as a primary product on December 05, 2014 (it was previously only a cross-sell item). Could you please pull sales data since then, and show how well each product cross-sells from one another?

```

CREATE TEMPORARY TABLE primary_products
SELECT
  order_id,
  primary_product_id,
  created_at
FROM orders
WHERE created_at > '2014-12-04';

SELECT
  primary_product_id,
  COUNT(DISTINCT order_id) AS total_orders,
  COUNT(CASE WHEN second_pro = 1 THEN order_id ELSE NULL END) AS xsold_p1,
  COUNT(CASE WHEN second_pro = 2 THEN order_id ELSE NULL END) AS xsold_p2,
  COUNT(CASE WHEN second_pro = 3 THEN order_id ELSE NULL END) AS xsold_p3,
  COUNT(CASE WHEN second_pro = 4 THEN order_id ELSE NULL END) AS xsold_p4,
  COUNT(CASE WHEN second_pro = 1 THEN order_id ELSE NULL END)
  /COUNT(DISTINCT order_id) AS xsold_p1_conv_rt,
  COUNT(CASE WHEN second_pro = 2 THEN order_id ELSE NULL END)
  /COUNT(DISTINCT order_id) AS xsold_p2_conv_rt,
  COUNT(CASE WHEN second_pro = 3 THEN order_id ELSE NULL END)
  /COUNT(DISTINCT order_id) AS xsold_p3_conv_rt,
  COUNT(CASE WHEN second_pro = 4 THEN order_id ELSE NULL END)
  /COUNT(DISTINCT order_id) AS xsold_p4_conv_rt
FROM
(
  SELECT
    primary_products.*,
    order_items.product_id AS second_pro
  FROM primary_products
  LEFT JOIN order_items
    ON primary_products.order_id = order_items.order_id
    AND is_primary_item = 0
) AS second_pro
GROUP BY 1

```

	primary_product_id	total_orders	xsold_p1	xsold_p2	xsold_p3	xsold_p4	xsold_p1_conv_rt	xsold_p2_conv_rt	xsold_p3_conv_rt	xsold_p4_conv_rt
▶	1	4509	0	241	555	942	0.0000	0.0534	0.1231	0.2089
	2	1285	25	0	40	263	0.0195	0.0000	0.0311	0.2047
	3	939	86	40	0	211	0.0916	0.0426	0.0000	0.2247
	4	581	16	9	22	0	0.0275	0.0155	0.0379	0.0000