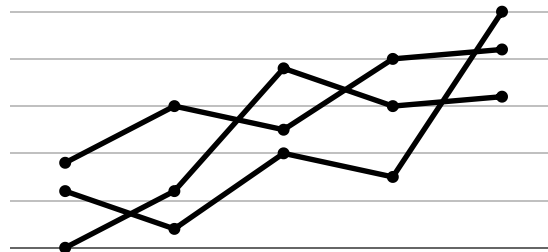


# ADVANCED E-COMMERCE ANALYTICS

TELLING BUSINESS STORY WITH NUMBERS USING TRENDED  
PERFORMANCE DATA

The goal of this project is to help the company secure a new round of funding by telling a compelling data driven story about the company's growth over the past three years in the ecommerce business.



The tools used in this project:

- MYSQL Workbench
- Tableau Desktop
- Microsoft Excel
- Report Design Template

Solution, Design and Report done by:  
Rachid El Khayat

# TABLE OF CONTENTS

---

## **01** INTRODUCTION

---

## **02** THE PROJECT REQUIREMENTS

---

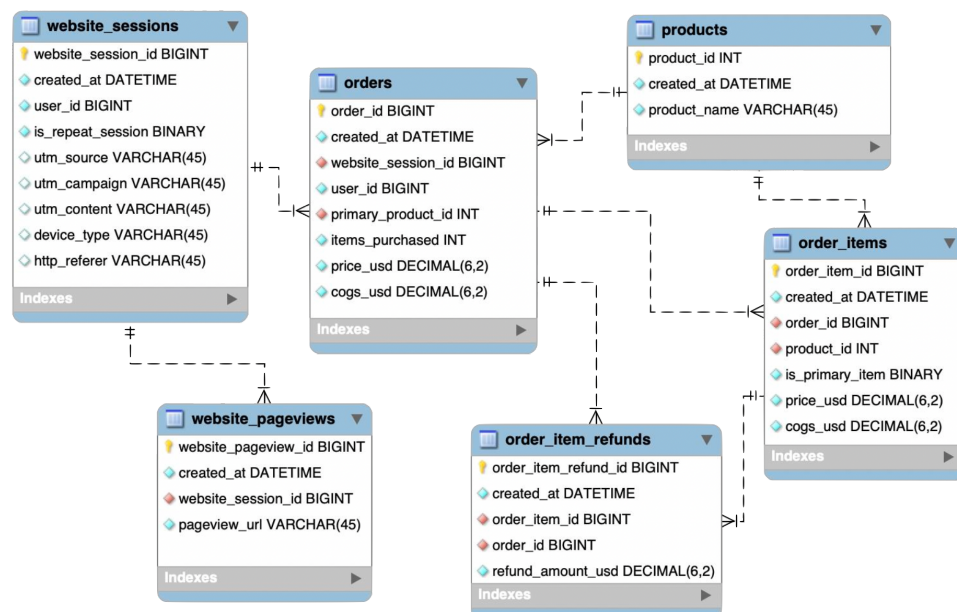
## **03** The Solution

# INTRODUCTION

This project is a part of the advanced SQL course that was focused on simulating a real life analytics challenges. The two key take aways from this course are:

- The detailed e-commerce knowledge, with all what it brings from statistics and understanding the anatomy of this industry.
- The advanced technical part, the cases go in depth in all the advanced sql concepts i.e Multiple joins, pivoting, aggregating, time series analysis,... .

The database used in this course is based on a dummy database provided by the course creator.



We will be working with six related tables, which contain E-Commerce data about:

- the 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.

## THE PROJECT REQUIREMENTS

This project is meant to show the coding a visualizing skills by telling the story of growth of the company using different data analysis techniques. the main metrics of success of this project are:

### Growth

Exponential Growth as shown by numbers

### Sales Channels

different sales channels

### Cross selling

adding new products helps with the overall revenue

This project is meant to show the coding a visualizing skills by telling the story of growth of the company using different data analysis techniques.

# THE SOLUTION

Presenting the company volume growth trend for the life of the business

## SQL SCRIPT

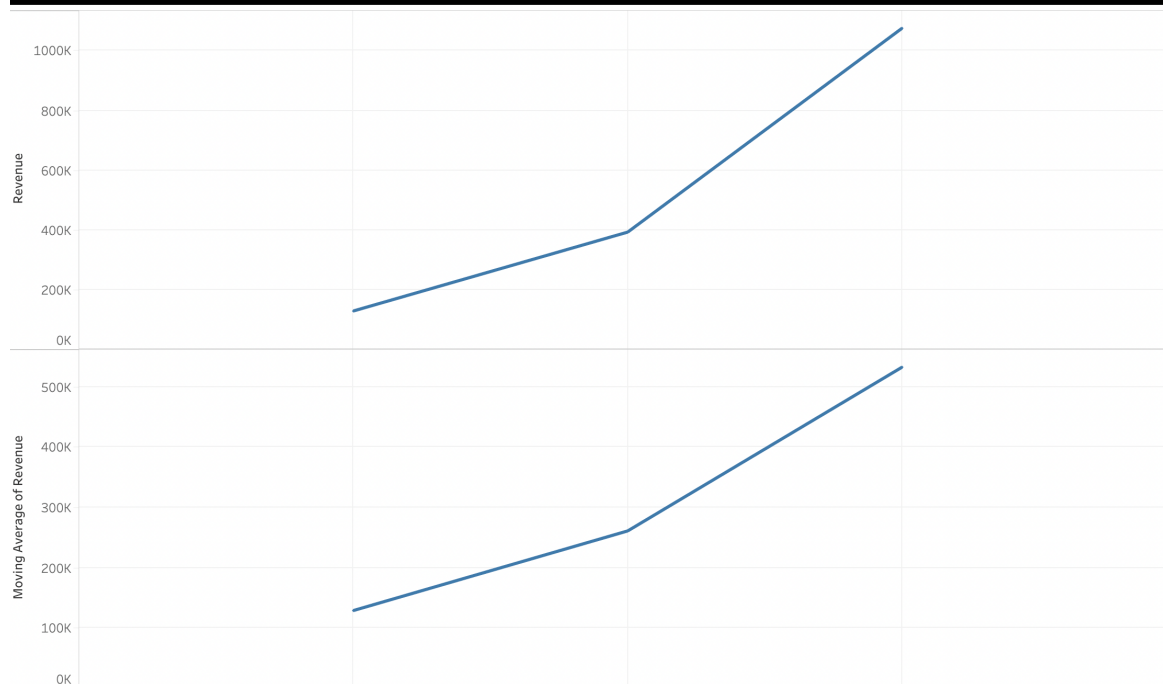
```
select
    year(A.created_at)                as Year
    ,quarter(A.created_at)            as Quarter
    ,count(distinct A.website_session_id) as Sessions
    ,count(distinct A1.order_id)       as Orders
    ,sum(A1.price_usd)                 as Revenue

    ,count(distinct A1.order_id)/count(distinct A.website_session_id) as CVR
    ,round(sum(A1.price_usd) /count(distinct A1.order_id),2)
    as Rev_per_order
    ,round(sum(A1.price_usd)/count(distinct A.website_session_id),2)
    as Rev_per_session

from website_sessions A
left join orders A1
on A.website_session_id = A1.website_session_id
where year(A.created_at) <= 2014

group by 1,2
```

Overall growth through out the operation period



Presenting the company volume growth, trended by quarter for the life of the business

#### SQL SCRIPT

```
with cte1 as (
select
    B.*,
    B1.order_id,

    case
    when utm_source='gsearch' and utm_campaign ='nonbrand'    then 'gsearch'
    when utm_source='bsearch' and utm_campaign ='nonbrand'    then 'bsearch'
    when utm_campaign ='brand'                                then 'brand'
    when utm_source is null and http_referer is not null       then 'organic'
    when utm_source is null and http_referer is null           then 'direct'
    end as channels

from website_sessions B
left join orders    B1
on B.website_session_id = B1.website_session_id
)

select

year(created_at) as yr
,quarter(created_at) as qrt

,count( distinct case when channels='gsearch' then order_id end) as gsearch
,count( distinct case when channels='gsearch' then order_id end)/
    count( distinct case when channels='gsearch' then website_session_id end) as
gsearch_CVR

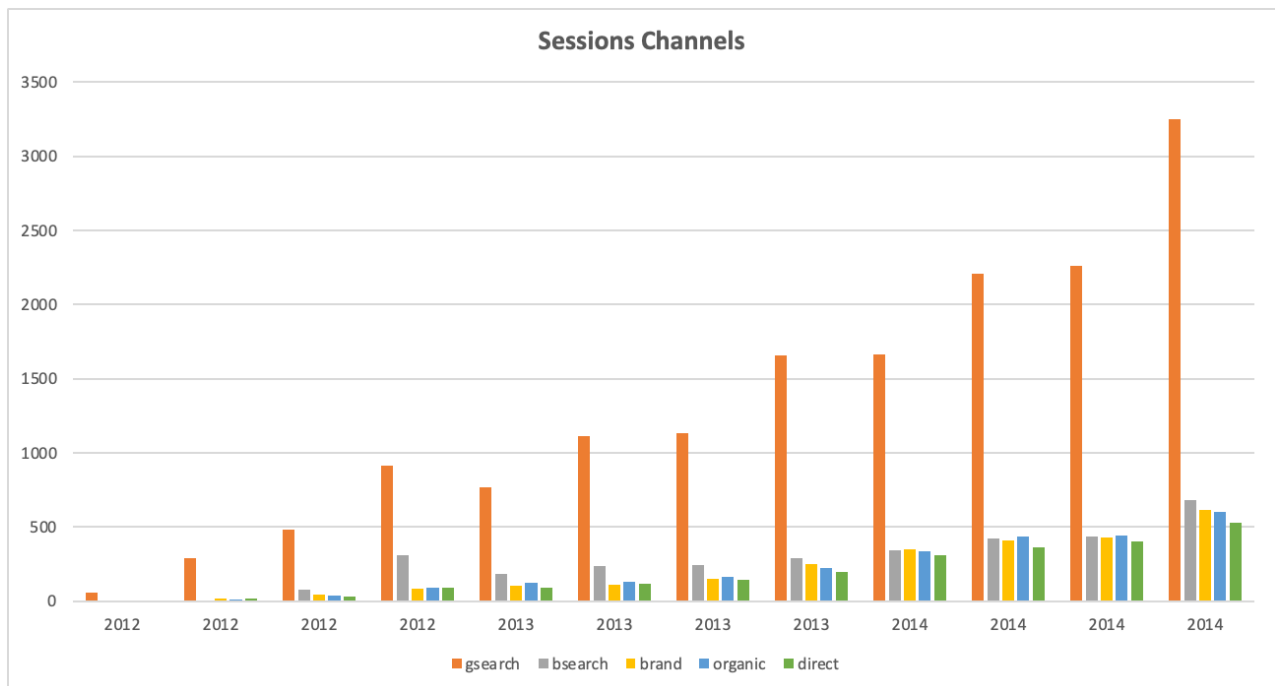
,count( distinct case when channels='bsearch' then order_id end) as bsearch
,count( distinct case when channels='bsearch' then order_id end)/
    count( distinct case when channels='bsearch' then website_session_id end) as
bsearch_CVR

,count( distinct case when channels='brand' then order_id end) as brand
,count( distinct case when channels='brand' then order_id end)/
    count( distinct case when channels='brand' then website_session_id end) as
brand_CVR
```

```
,count( distinct case when channels='organic' then order_id end) as organic
,count( distinct case when channels='organic' then order_id end)/
count( distinct case when channels='organic' then website_session_id end) as
organic_CVR
```

```
,count( distinct case when channels='direct' then order_id end) as direct
,count( distinct case when channels='direct' then order_id end)/
count( distinct case when channels='organic' then website_session_id end) as
direct_CVR
```

```
from cte1
group by 1,2
order by 1,2
```



Clarifying the effect of the diversification of the products in different price range and its effect on the revenue over time.

#### SQL SCRIPT

select

```
year(created_at) as yr  
,month(created_at) as mo
```

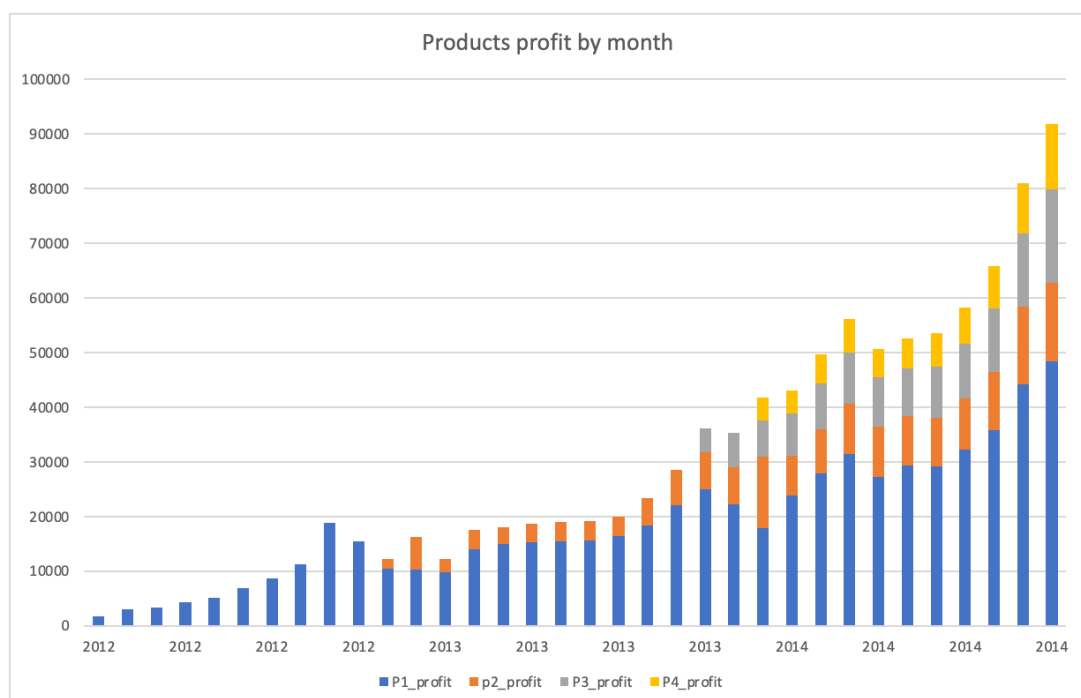
```
,sum(price_usd) as rev  
,sum(price_usd) - sum(cogs_usd) as profit
```

```
,sum(case when product_id= 1 then price_usd end ) as p1_rev  
,sum(case when product_id= 1 then price_usd end ) - sum(case when  
product_id= 1 then cogs_usd end ) as P1_profit  
,sum(case when product_id= 2 then price_usd end ) as p2_rev  
,sum(case when product_id= 2 then price_usd end ) - sum(case when  
product_id= 2 then cogs_usd end ) as p2_profit  
,sum(case when product_id= 3 then price_usd end ) as p3_rev  
,sum(case when product_id= 3 then price_usd end ) - sum(case when  
product_id= 3 then cogs_usd end ) as P3_profit  
,sum(case when product_id= 4 then price_usd end ) as p4_rev  
,sum(case when product_id= 4 then price_usd end ) - sum(case when  
product_id= 4 then cogs_usd end ) as P4_profit
```

from order\_items

group by 1,2

order by 1,2





Showing the clickthrough rate from different products into orders.

#### SQL SCRIPT

```
with products_pageviews as(

select *
from website_pageviews
where pageview_url = '/products'

),
next_page_after_product_pageviews as( -- getting the next page after the
product

select
s.website_session_id
,min(s.website_pageview_id) as next_page
,s1.pageview_url as next_page_url

from products_pageviews S
left join website_pageviews S1
on s1.website_session_id = s.website_session_id
and s1.website_pageview_id > s.website_pageview_id

where s1.pageview_url is not null

group by 1
)
select
year(c.created_at) as yr
,month(c.created_at) as mo
,count(distinct c.website_session_id) as total_sessions
,count(distinct c1.website_session_id) as products_sessions
,count(distinct c2.website_session_id) as pages_after_product
,count(distinct c3.website_session_id) as orders

, count(distinct c1.website_session_id)/count(distinct c.website_session_id) as
CTR_sessions
, count(distinct c2.website_session_id)/count(distinct c1.website_session_id) as
CTR_product
, count(distinct c3.website_session_id)/count(distinct c1.website_session_id) as
CVR_product_to_orders
```

```

from website_pageviews C
left join products_pageviews C1
on c.website_session_id =C1.website_session_id
left join next_page_after_product_pageviews C2
on c.website_session_id =C2.website_session_id
left join orders C3
on c.website_session_id =C3.website_session_id

```

```

group by 1,2
order by 1,2

```



products cross sales growth as different products with different price point where added, clients tend to buy additional product along with item1

#### SQL SCRIPT

```

with requested_orders as (
SELECT
order_id
,primary_product_id
,created_at as ordered_at

FROM orders
WHERE created_at > '2014-12-05'

), CS_items as(

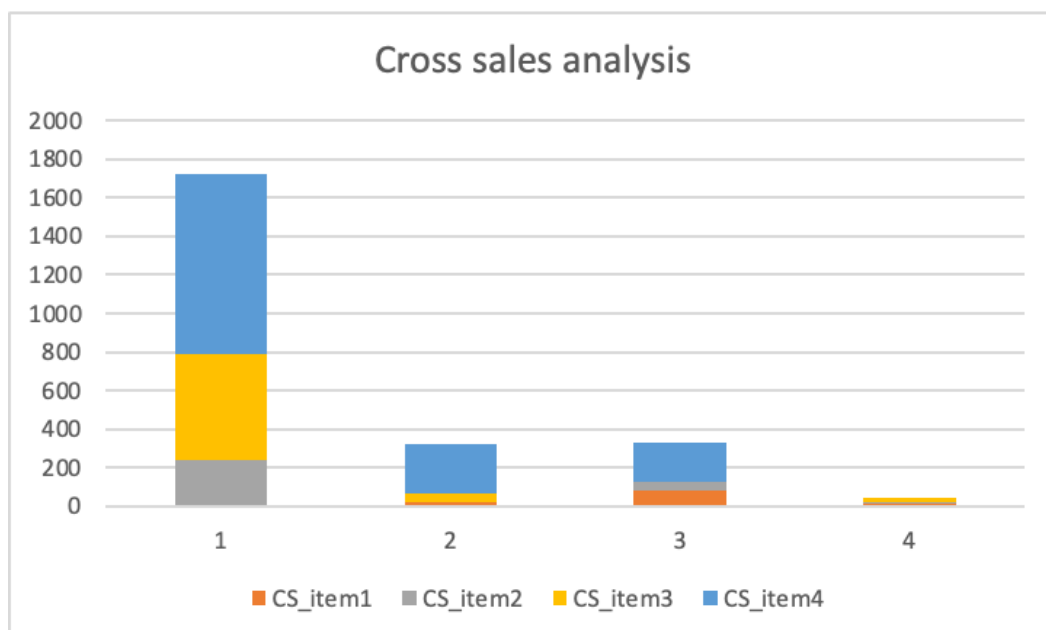
```

```
SELECT
D.*,
D1.product_id AS CS_item_id
```

```
FROM requested_orders D
LEFT JOIN order_items D1
ON D.order_id = D1.order_id
AND D1.is_primary_item = 0
)
```

```
select
primary_product_id
, COUNT(DISTINCT order_id) AS total_orders
, COUNT(DISTINCT CASE WHEN CS_item_id = 1 THEN order_id ELSE NULL END) AS
CS_item1
, COUNT(DISTINCT CASE WHEN CS_item_id = 2 THEN order_id ELSE NULL END) AS
CS_item2
, COUNT(DISTINCT CASE WHEN CS_item_id = 3 THEN order_id ELSE NULL END) AS
CS_item3
, COUNT(DISTINCT CASE WHEN CS_item_id = 4 THEN order_id ELSE NULL END) AS
CS_item4
, COUNT(DISTINCT CASE WHEN CS_item_id = 1 THEN order_id ELSE NULL
END)/COUNT(DISTINCT order_id) AS CS_item1_rate
, COUNT(DISTINCT CASE WHEN CS_item_id = 2 THEN order_id ELSE NULL
END)/COUNT(DISTINCT order_id) AS CS_item1_rate
, COUNT(DISTINCT CASE WHEN CS_item_id = 3 THEN order_id ELSE NULL
END)/COUNT(DISTINCT order_id) AS CS_item1_rate
, COUNT(DISTINCT CASE WHEN CS_item_id = 4 THEN order_id ELSE NULL
END)/COUNT(DISTINCT order_id) AS CS_item1_rate

from CS_items
group by 1
```



---

***Thank you***

---