

# **SQL BUSINESS CASE**

**¿0 morirás en el intento?**

# BUSINESS CASE

## Analizando Productos de Amazon Consumer Electronics

Eres un Analista de Datos en Amazon.com (AMZN). El equipo de Consumer Electronics (CE) te ha solicitado que los ayudes a entender cómo es el posicionamiento de AMZN vs. otros retailers.

### Detalles

Data disponible publicamente (*scrapped*) en 2017/2018 para productos en Amazon.com

- Table 1: *product\_data*; contiene atributos de productos
- Table 2: *amazon*; incluye precios de productos para Amazon
- Table 3: *competitors*; contiene precios de competidores

Supuestos,

- la tabla *product\_data* contiene **todos** los productos de CE en Amazon.com
- la tabla *competitors* contiene **todos** los competidores

Recomendaciones,

- utilizar *number\_of\_reviews* como variable proxy para ventas
- utilizar tantas queries como sean necesarias

Enviar a ***julian@dragonlab.team***:

- SQL relevante utilizado para cada ejercicio

# BUSINESS CASE

## EX 1: Identificando competidores más relevantes

¿Qué competidores deberíamos mirar más de cerca? Buscamos identificar a aquellos competidores que venden los mismos productos que nosotros.

Identificar top 4 competidores, según porcentaje de catálogo coincidente con de Amazon.

| Ranking | Competidor       | Nro ASINs | % ASINs sobre total en Amazon |
|---------|------------------|-----------|-------------------------------|
| 1       | Bestbuy.com      | 812       | 84%                           |
| 2       | bhphotovideo.com | 491       | 51%                           |
| 3       | Walmart.com      | 234       | 24%                           |
| 4       | Beach Camera     | 70        | 7%                            |

# BUSINESS CASE

## EX 1: Identificando competidores más relevantes

```
-- number of unique ASINs. Respuesta: 500
```

```
SELECT  
count(DISTINCT prod.asin) AS selection  
FROM product_data prod
```

```
-- coincidencia de la selección con competidores
```

```
WITH SUB AS(
```

```
SELECT DISTINCT  
comp.prices_merchant  
,comp.asin
```

```
FROM competitors comp  
INNER JOIN product_data prod  
ON prod.asin=comp.asin
```

```
)
```

```
SELECT  
prices_merchant  
,COUNT(DISTINCT SUB.ASIN) AS OVERLAP  
,round(COUNT(DISTINCT SUB.ASIN)*1.0/500,2) AS OVERLAP_PERC
```

```
FROM SUB
```

```
GROUP BY 1
```

```
ORDER BY OVERLAP DESC
```

# BUSINESS CASE

## EX 2: Identificando categorías más relevantes

¿Qué categorías de productos son las más importantes? Buscamos identificar a aquellas categorías de productos con mayor número de *reviews*.

Identificar top 5 categorías, según porcentaje de reviews.

| Ranking | Product Category    | Count of ASINs | Share in total ASINs | SUM of reviews | Share in total reviews |
|---------|---------------------|----------------|----------------------|----------------|------------------------|
| 1       | <b>pc</b>           | 269            | 28%                  | 93803          | 45%                    |
| 2       | <b>electronics</b>  | 374            | 39%                  | 66034          | 32%                    |
| 3       | <b>wireless</b>     | 122            | 13%                  | 21971          | 10%                    |
| 4       | camera              | 72             | 7%                   | 13930          | 7%                     |
| 5       | home_entertainment  | 81             | 8%                   | 11086          | 5%                     |
| 6       | musical_instruments | 12             | 1%                   | 1713           | 1%                     |

# BUSINESS CASE

## EX 2: Identificando categorías más relevantes

```
-- Total número de reviews acumuladas por todos los prods
-- Answer: 131280
SELECT
SUM(number_of_reviews) AS SUM_REVIEWS
FROM product_data
```

```
-- Cómo el total de reviews se distribuyen a lo largo de las categorías?

SELECT
product_category
,COUNT(DISTINCT asin) AS COUNT_ASINS
,ROUND(COUNT(DISTINCT asin)/500,2) AS SHARE_ASINS

,SUM(number_of_reviews) AS SUM_REVIEWS
,ROUND(SUM(number_of_reviews)/ 131280,2) AS SHARE_REVIEWS

FROM product_data

GROUP BY 1

ORDER BY SUM_REVIEWS DESC
```

# BUSINESS CASE

## EX 3: Adquisición de nuevos clientes

La tabla de ecommerce contiene datos de órdenes a lo largo de dos años (2010 y 2011). Extraer porcentaje de clientes nuevos mensuales a lo largo de 2011. *Nota: entendemos un cliente nuevo aquel que ha sido captado en 2011, es decir, que no ha comprado durante 2010.*

- *Tabla: ecommerce*

### *Output esperado*

| Row | year | month | total_shoppers | new_shoppers | existing_shoppers | new_shopper_rate |
|-----|------|-------|----------------|--------------|-------------------|------------------|
| 1   | 2011 | 1     | 784            | 421          | 363               | 0.54             |
| 2   | 2011 | 2     | 799            | 481          | 318               | 0.6              |
| 3   | 2011 | 3     | 1021           | 653          | 368               | 0.64             |
| 4   | 2011 | 4     | 900            | 558          | 342               | 0.62             |
| 5   | 2011 | 5     | 1080           | 703          | 377               | 0.65             |
| 6   | 2011 | 6     | 1052           | 691          | 361               | 0.66             |
| 7   | 2011 | 7     | 994            | 657          | 337               | 0.66             |
| 8   | 2011 | 8     | 981            | 644          | 337               | 0.66             |
| 9   | 2011 | 9     | 1303           | 928          | 375               | 0.71             |
| 10  | 2011 | 10    | 1426           | 1071         | 355               | 0.75             |
| 11  | 2011 | 11    | 1712           | 1237         | 475               | 0.72             |
| 12  | 2011 | 12    | 687            | 426          | 261               | 0.62             |



# BUSINESS CASE

## EX 3: Adquisición de nuevos clientes

```
-- nuevos clientes
WITH
BASE_2010 AS(
-- compradores únicos 2010
SELECT DISTINCT
CustomerID

FROM ecommerce_formatdate ecommerce
WHERE
EXTRACT(YEAR FROM InvoiceDate) = 2010
)

,SUBQUERY AS(
-- compradores únicos 2011
SELECT
EXTRACT(YEAR FROM left_table.InvoiceDate) AS year
,EXTRACT(MONTH FROM left_table.InvoiceDate) AS month
,COUNT(DISTINCT left_table.CustomerID) AS total_shoppers
,COUNT(DISTINCT CASE WHEN right_table.CustomerID IS NULL THEN left_table.CustomerID END) AS new_shoppers
,COUNT(DISTINCT CASE WHEN right_table.CustomerID IS NOT NULL THEN left_table.CustomerID END) AS existing_shoppers

FROM ecommerce_formatdate left_table
LEFT JOIN BASE_2010 right_table
ON left_table.CustomerID = right_table.CustomerID

WHERE
EXTRACT(YEAR FROM left_table.InvoiceDate) = 2011

GROUP BY 1,2
)

SELECT
*
,ROUND(new_shoppers/total_shoppers,2) AS new_shopper_rate

FROM SUBQUERY

ORDER BY year,month
```