



# GOURMET

## Uso de MySQL y MongoDB en un caso de negocio

BEDU

#data-analysis-gdl-20-03

Módulo 1: introducción a  
bases de datos

Andrés Cervantes

# Agenda

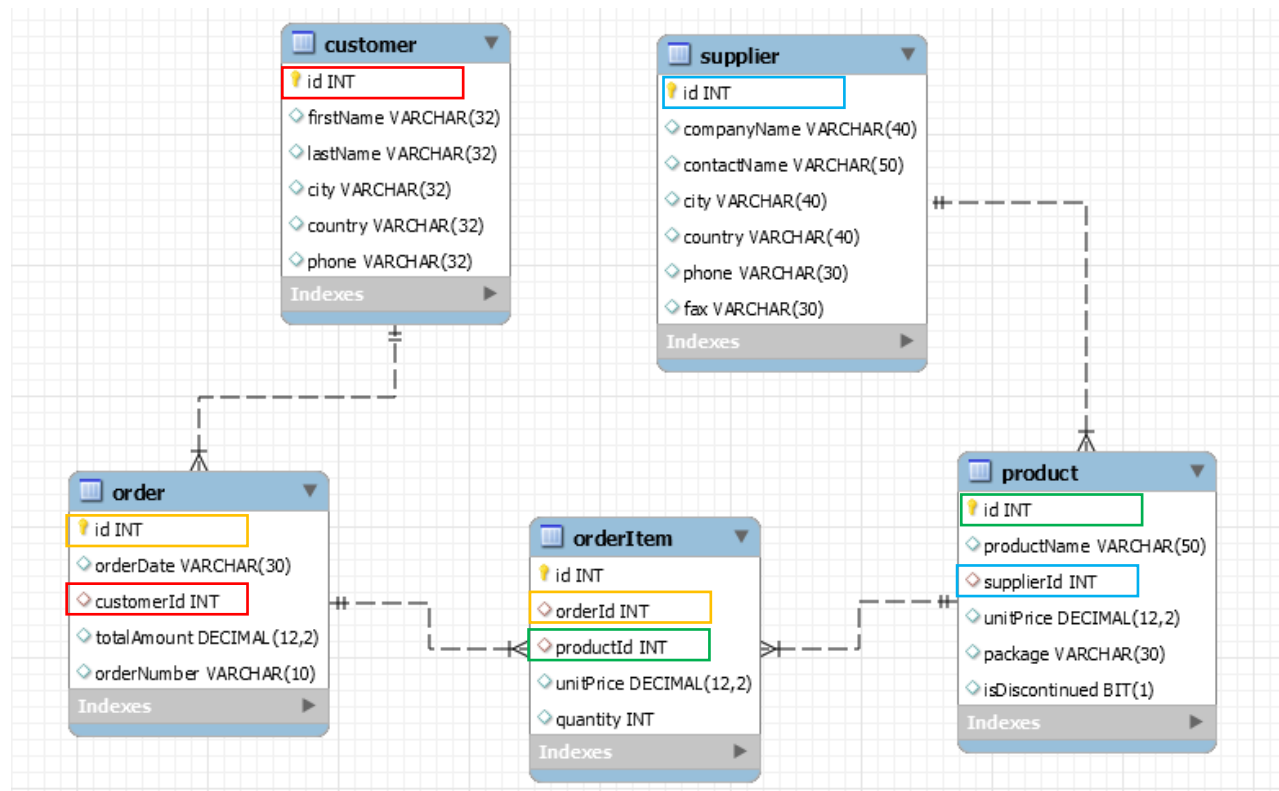
1. Objetivo
2. Contexto
3. Problema
4. Solución
5. Conclusiones

# 1. Objetivo

- Demostrar el manejo de MySQL y MongoDB
  - Estructura de BDD definida, uso adecuado de tipos de datos, buenas prácticas
  - Registros/documentos que se adecúan a la problemática
  - Consultas sencillas y elaboradas
  - Empleo de vistas

## 2. Contexto (ficticio)

- Manejo una tienda que comercializa productos de comida gourmet.



Nota: los datos utilizados se obtuvieron de [DoFactory](https://dofactory.com/)

# Problema

- Es marzo del 2020 y la pandemia ha llegado a México. Se prevee una recesión económica por lo que es necesario reducir costos
- Suponiendo que al reducir mi catálogo se reducen mis costos de manera proporcional, ¿qué medidas puedo tomar?

# Solución: Identificar a mis mejores clientes y qué compran.

- A. Identificar aquellas órdenes que hacen el 20% de mi venta.
- B. Identificar a los clientes de esas órdenes.
- C. Y cuáles son sus productos favoritos.

Nota: la solución es sólo un ejemplo para practicar *queries* sin fundamentos de negocios.

# A. Identificar aquellas órdenes que hacen el 20% de mi venta.



## MySQL

```
select sum(totalAmount)
from `order`;
```

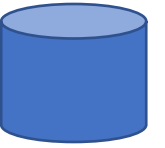
```
select *
from `order`
order by totalAmount
desc;
```

```
select *, (totalAmount / 1354448.59) as relativeTotalAmount
from `order`
order by totalAmount desc
limit 28;
```

## MongoDB

```
[
  { $sort: { totalAmount: -1}},
  { $addFields: {
    relativeTotalAmount: {$divide: ['$totalAmount', 1354458.59]}}}},
  { $limit: 28},
  { $group: {
    _id: null,
    topOrders: {$sum: '$relativeTotalAmount'}
  }}
]
```

# A. Identificar aquellas órdenes que hacen el 20% de mi venta.



## MySQL

id	orderDate	customerId	totalAmount	orderNumber	relativeTotalAmount
618	Feb 2 2014 12:00:00:000AM	63	17250.00	542995	0.012736
783	Apr 17 2014 12:00:00:000AM	71	16321.90	543160	0.012051
734	Mar 27 2014 12:00:00:000AM	34	15810.00	543111	0.011673
125	Dec 4 2012 12:00:00:000AM	62	12281.20	542502	0.009067
177	Jan 23 2013 12:00:00:000AM	51	11493.20	542554	0.008486
570	Jan 6 2014 12:00:00:000AM	39	11490.70	542947	0.008484
642	Feb 16 2014 12:00:00:000AM	65	11380.00	543019	0.008402

	cumulativeRelativeTotalAmount
▶	0.200541

## MongoDB

```
_id: ObjectId("5ef5970b829e972ae85d0e0b")
id: 618
orderDate: "Feb 2 2014 12:00:00:000AM"
customerId: 63
totalAmount: 17250
orderNumber: 542995
relativeTotalAmount: 0.01273571604725102743820318641118441
```

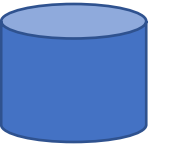
```
id: 783
orderDate: "Apr 17 2014 12:00:00:000AM"
customerId: 71
totalAmount: 16321.9
orderNumber: 543160
relativeTotalAmount: 0.01205049760878994462281788917592
```

```
id: 734
orderDate: "Mar 27 2014 12:00:00:000AM"
customerId: 34
totalAmount: 15810
orderNumber: 543111
relativeTotalAmount: 0.01167256062069789819118796389338
```

```
_id: null
cumulativeTopOrders: 0.20053642245349117686942352368262
```



# B. Identificar a los clientes de las mejores órdenes.



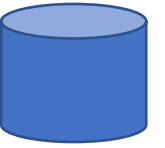
## MySQL

```
SELECT *
FROM customer
WHERE id IN (
    SELECT customerId
    FROM (
        SELECT customerId
        FROM `order`
        ORDER BY totalAmount DESC
        LIMIT 28) AS o
    GROUP BY customerId);
```

## MongoDB

```
[{$sort:
    { totalAmount: -1}},
{$addFields:
    { relativeTotalAmount:
        {$divide:['$totalAmount',1354458.59]}}, {$limit: 28},
{$project:
    { _id:0,
      customerId:1}},
{$lookup:
    { from: 'customer',
      localField: 'customerId',
      foreignField: 'Id',
      as: 'customer'}
}]
```

# B. Identificar a los clientes de las mejores órdenes.



## MySQL

id	firstName	lastName	city	country	phone
63	Horst	Kloss	Cunewalde	Germany	0372-035188
71	Jose	Pavarotti	Boise	USA	(208) 555-8097
34	Mario	Pontes	Rio de Janeiro	Brazil	(21) 555-0091
62	Lúcia	Carvalho	Sao Paulo	Brazil	(11) 555-1189
51	Jean	Fresnière	Montréal	Canada	(514) 555-8054
39	Philip	Cramer	Brandenburg	Germany	0555-09876
65	Paula	Wilson	Albuquerque	USA	(505) 555-5939

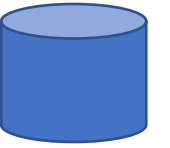
```
count(*)  
15
```

## MongoDB

```
customerId: 63  
customer: Array  
  0: Object  
    _id: ObjectId("5ef5977f829e972ae85d0f1e")  
    Id: 63  
    FirstName: "Horst"  
    LastName: "Kloss"  
    City: "Cunewalde"  
    Country: "Germany"  
customerId: 71  
customer: Array  
  0: Object  
    _id: ObjectId("5ef5977f829e972ae85d0f26")  
    Id: 71  
    FirstName: "Jose"  
    LastName: "Pavarotti"  
    City: "Boise"  
    Country: "USA"
```

```
_id: null  
countTopCustomer: 15
```

# C. Cuáles son los productos favoritos de mis mejores clientes.

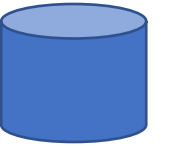


## MySQL

```
select p.id as productId, productName, supplierId,  
p.unitPrice, package, isDiscontinued, count(*) as  
popularity  
from product as p  
join orderItem as oi  
    on p.id = oi.productId  
join `order` as o  
    on oi.orderId = o.id  
where customerId in (  
    select id  
    from topCustomer)  
group by productId order by popularity desc;
```

## MongoDB

# C. Cuáles son los productos favoritos de mis mejores clientes.



## MySQL

productId	productName	supplierId	unitPrice	package	isDiscontinued	popularity
31	Gorgonzola Telino	14	12.50	12 - 100 g pkgs	0	24
62	Tarte au sucre	29	49.30	48 pies	0	21
59	Raclette Courdavault	28	55.00	5 kg pkg.	0	20
56	Gnocchi di nonna Alice	26	38.00	24 - 250 g pkgs.	0	20
2	Chang	1	19.00	24 - 12 oz bottles	0	19

## MongoDB

# Conclusiones

1. Entendí mejor MySQL que MongoDB (estructura, diagramas, código) pero MongoDB parece ser una herramienta más poderosa. Hay que practicar las agregaciones.
2. Los lenguajes de programación sin igual que las lenguas, se puede expresar lo mismo pero necesitas usar “gramática” diferente.
3. Se pueden hacer análisis muy interesantes (con pocos datos) complementando con estadística.
4. La parte más tardada fue encontrar los datos en sí, más que su procesamiento.