**Introducción a Bases de Datos 2020**

**\_\_\_**

**“Breve interpretación de los procesos de bases de datos y sus relaciones”**

**Checkpoint**

**Alumno:** Céspedes Jaén Abdiel

**Experto:** Ramírez Andrés

**Grupo:** data - analysis - gdl - 20 - 06

**Fecha de entrega:** Sábado 31/10/2020

**ÍNDICE**

[**Objetivos**](#_mv8q5knshei2) **3**

[Objetivo del curso](#_fjfowr1sxk29) 3

[Objetivo del proyecto](#_1v4bd5cvu8m8) 3

[**Contexto**](#_k2zdx4ehfy4f) **3**

[**ERR (Diagrama Entidad - Relación)**](#_44vg4ol3seg) **4**

[**Carga de Bases de Datos**](#_t95pub2uq4qj) **4**

[MySQL Workbench](#_lq9m77hjuxue) 4

[Mongodb Compass](#_47idxh7hi7ut) 5

[**Solución**](#_755aheicybil) **5**

[Plan de Solución](#_yxx60lgdo5b8) 5

[Procedimientos](#_1z5et5gjo23a) 6

[Exploración y consultas iniciales (Post Work I & IV)](#_5w27nfg4ycbk) 6

[Agrupamientos y segmentaciones (Post Work II & V)](#_ygo6tojldx80) 10

[Relaciones entre tablas (Post Work III & VI)](#_ugpd1zr1eibd) 17

[Resolución del problema](#_64m2vda381ky) 24

[**Conclusión**](#_5fsi9sfyksfe) **25**

[**Crédito a la Base de Datos**](#_er5kbiavqr8w) **25**

[Base de Datos Original](#_crtkzli9pa1r) 25

[Base de Datos Utilizada](#_clij5ol571oz) 25

# 

# Objetivos

## Objetivo del curso

Al finalizar el módulo serás capaz de utilizar los principales conceptos de las bases de datos mediante la generación de consultas en sistemas gestores de bases de datos relacionales y no relacionales.

## Objetivo del proyecto

La empresa, para poder brindar un mejor servicio, para evaluar los hábitos de compra de sus clientes y para generar una mejor experiencia con los usuarios, desea saber lo siguiente:

* ¿Qué clientes ya han sido satisfechos al haber registrado su estatus como entregado?
* ¿Cuál es el método de pago preferido de los usuarios?
* ¿Cuáles son los pagos que están por encima del promedio de su propio valor?
* ¿Cuál es el valor de pago de cada orden?

# Contexto

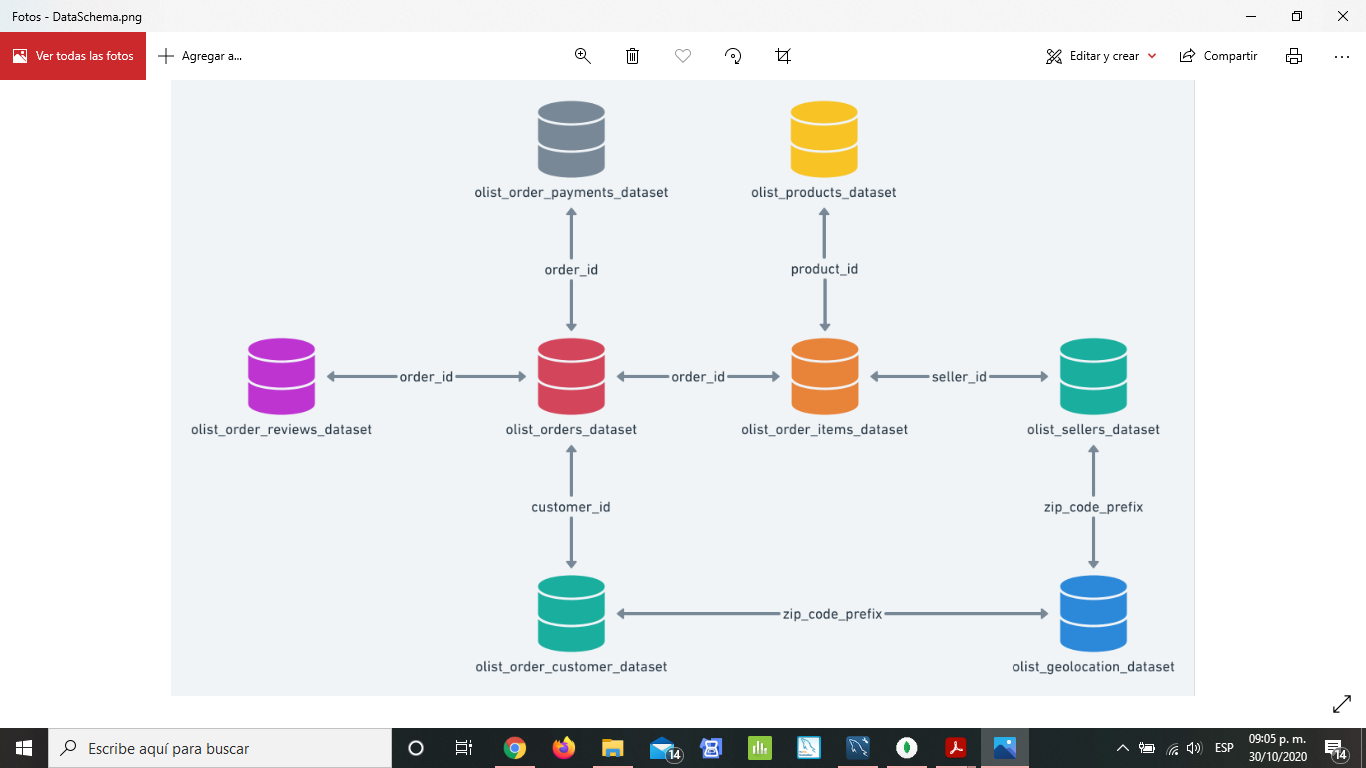
El presente trabajo fue realizado en base a una base de datos de una empresa brasileña de ecommerce, publicada por cortesía de Olist. Sus características permiten la visualización de su información a través de múltiples dimensiones: desde el estatus de la orden, el precio, el pago y el rendimiento del flete, la locación del cliente, atributos del producto y reseñas hechas por los propios clientes. La base de datos incluye también una tabla enfocada a la geolocalización, sustentada gracias a códigos postales.

Olist fue el proveedor de la base de datos empleada, el cual es el mayor departamento de almacenes en los mercados de Brasil. Olist conecta pequeños negocios a lo largo de Brasil a canales con un contrato sencillo. Los comerciantes pueden vender sus productos en las tiendas de Olist y enviarlas directamente a los clientes usando los socios de logística de Olist.

Una vez que al cliente se le hace llegar su producto, también se le envía una encuesta vía email para dar su reseña con respecto al proceso de compra.

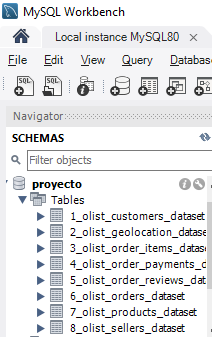
La base de datos, que está pensada para consultarse de modo relacional (SQL), será, además, tratada de manera no relacional (No SQL).

# ERR (Diagrama Entidad - Relación)

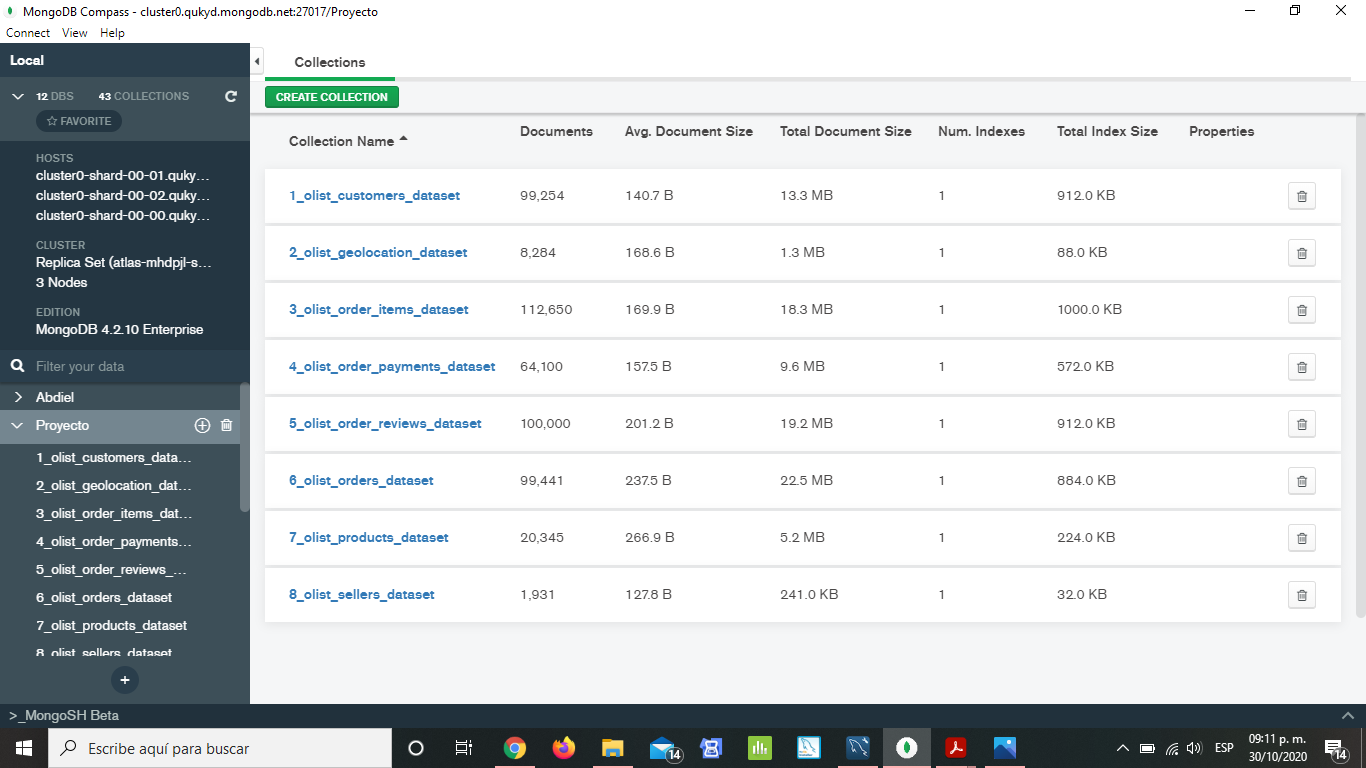


# Carga de Bases de Datos

## MySQL Workbench



## Mongodb Compass



# Solución

## Plan de Solución

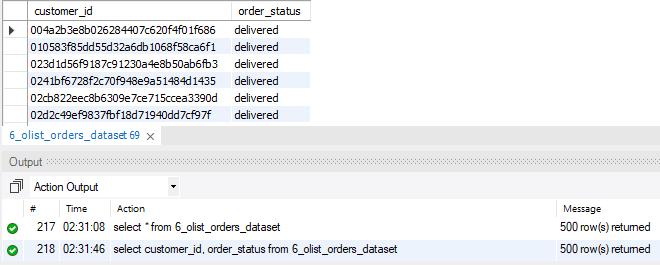
|  |  |
| --- | --- |
| **Post Work** | **Función** |
| 1 | Exploración y familiarización con las bases de datos para realizar consultas sencillas |
| 2 | Aporta la capacidad de segmentar inmensas cantidades de información de acuerdo con los criterios que más nos convengan |
| 3 | Además de aprender a configurar los propios servidores en la computadora para importar bases de datos, también permite aprender a relacionar estas y poder automatizar ciertas funciones |

## Procedimientos

### Exploración y consultas iniciales (Post Work I & IV)

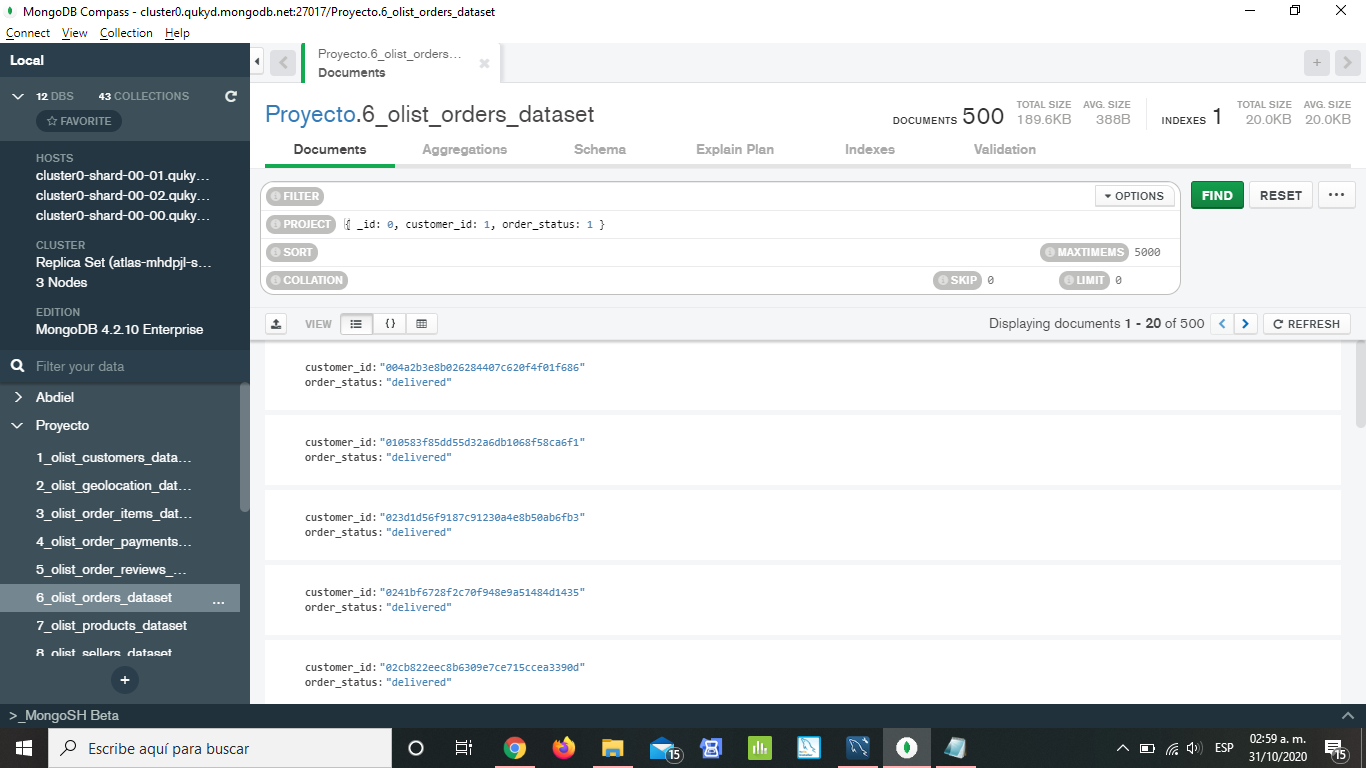
1) PROYECCIÓN. ¿Cuál es el estatus de la orden de cada cliente? (6)

MySQL WORKBENCH



SENTENCIA: select customer\_id, order\_status from 6\_olist\_orders\_dataset;

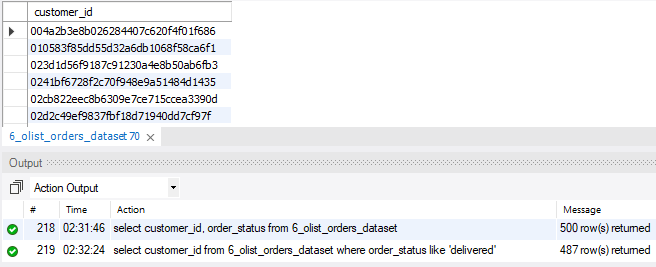
MONGODB



JSON: Project: { \_id: 0, customer\_id: 1, order\_status: 1 }

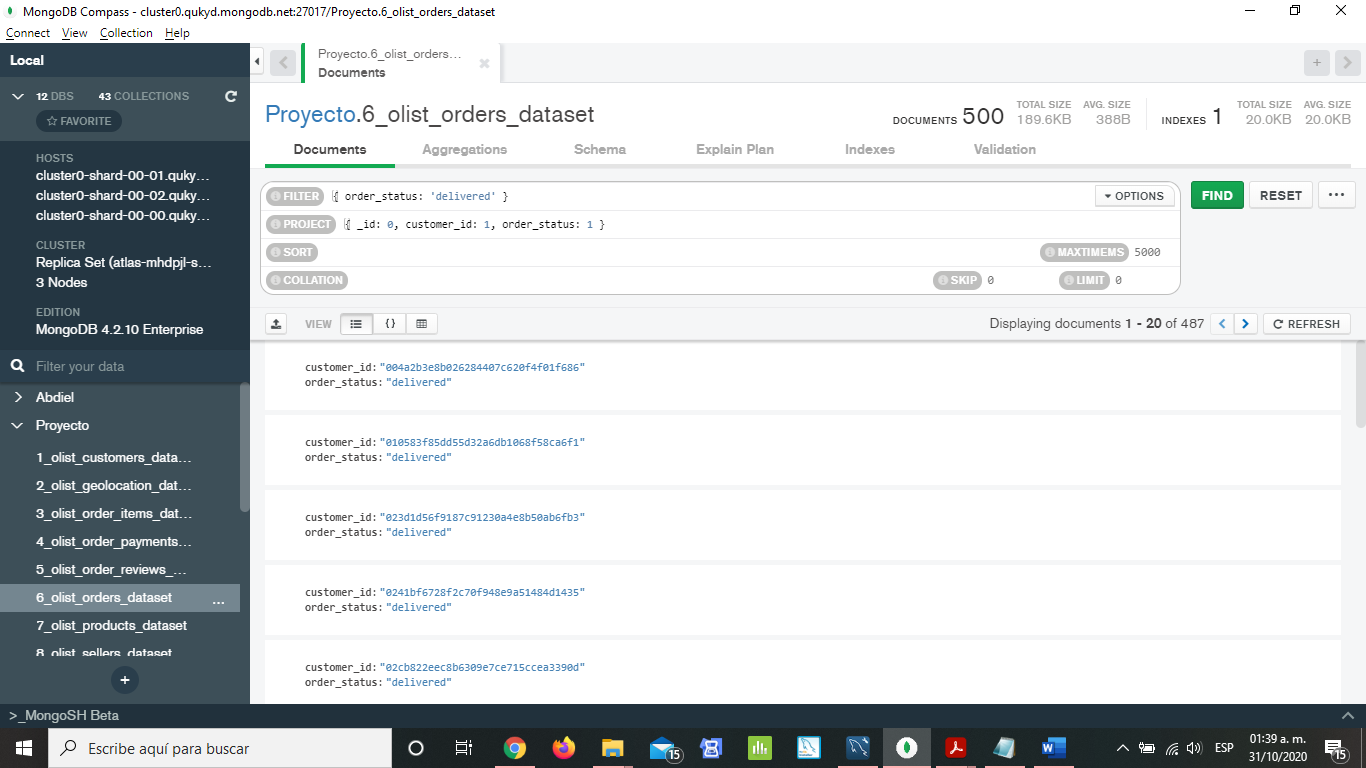
**2) FILTRO. ¿Qué clientes tienen un estatus de orden de “entregado”? (6)**

MySQL WORKBENCH



SENTENCIA: select customer\_id from 6\_olist\_orders\_dataset where order\_status like 'delivered';

MONGODB

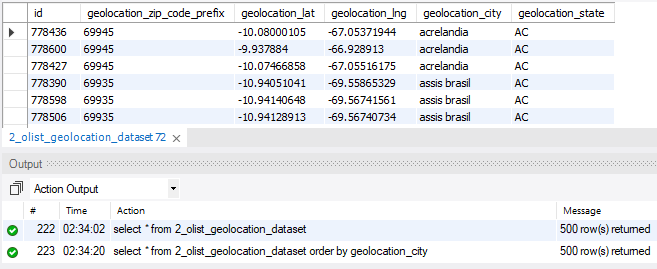


JSON: Filter: { order\_status: 'delivered' }

Project: { \_id: 0, customer\_id: 1, order\_status: 1 }

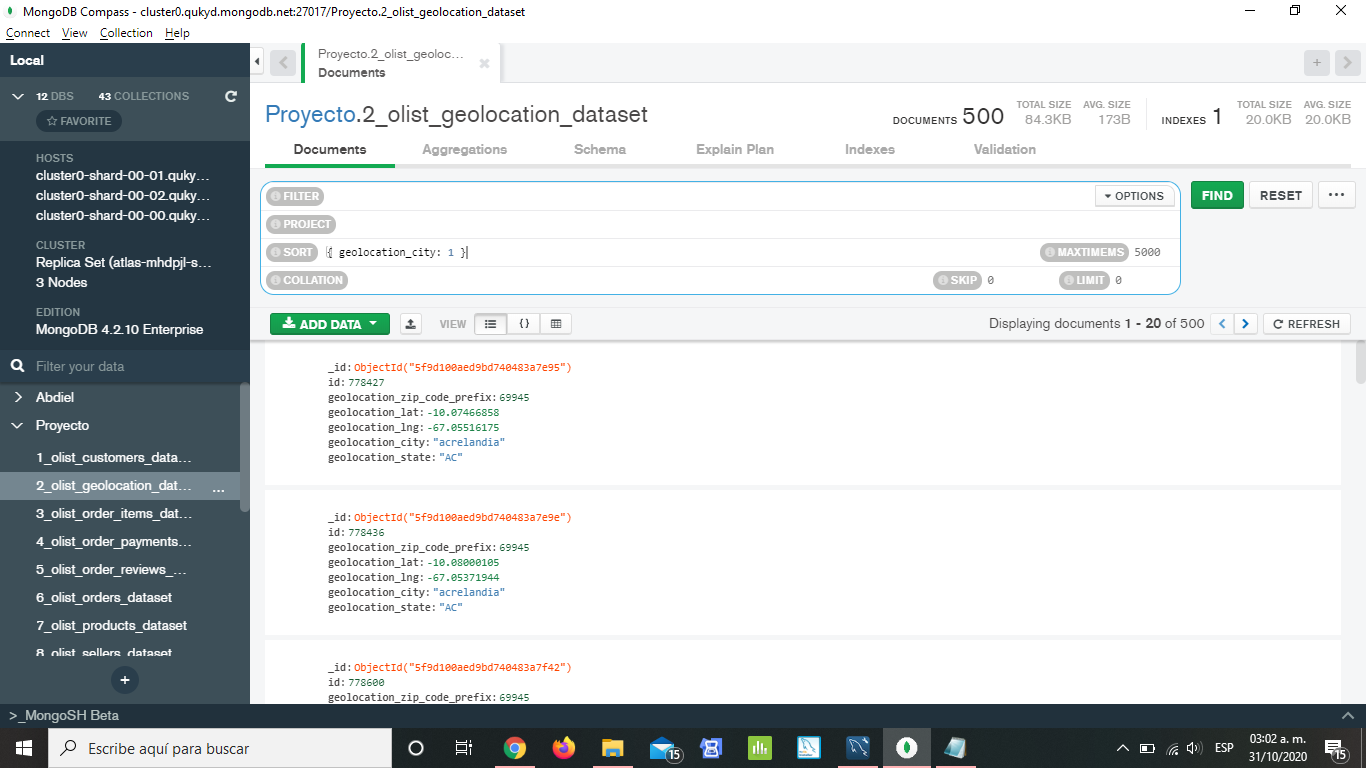
3) ORDENAMIENTO. ¿Cuál es el orden de los datos de la tabla de geolocalización por ciudad de manera ascendente? (2)

MySQL WORKBENCH



SENTENCIA: select \* from 2\_olist\_geolocation\_dataset order by geolocation\_city;

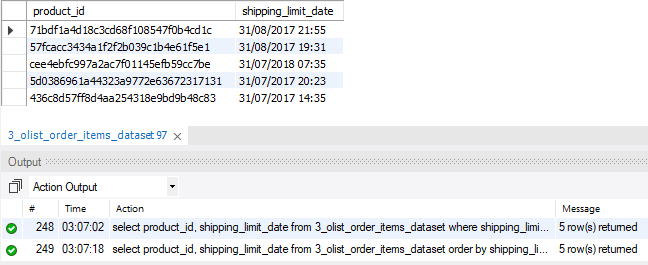
MONGODB



JSON: Sort: { geolocation\_city: 1 }

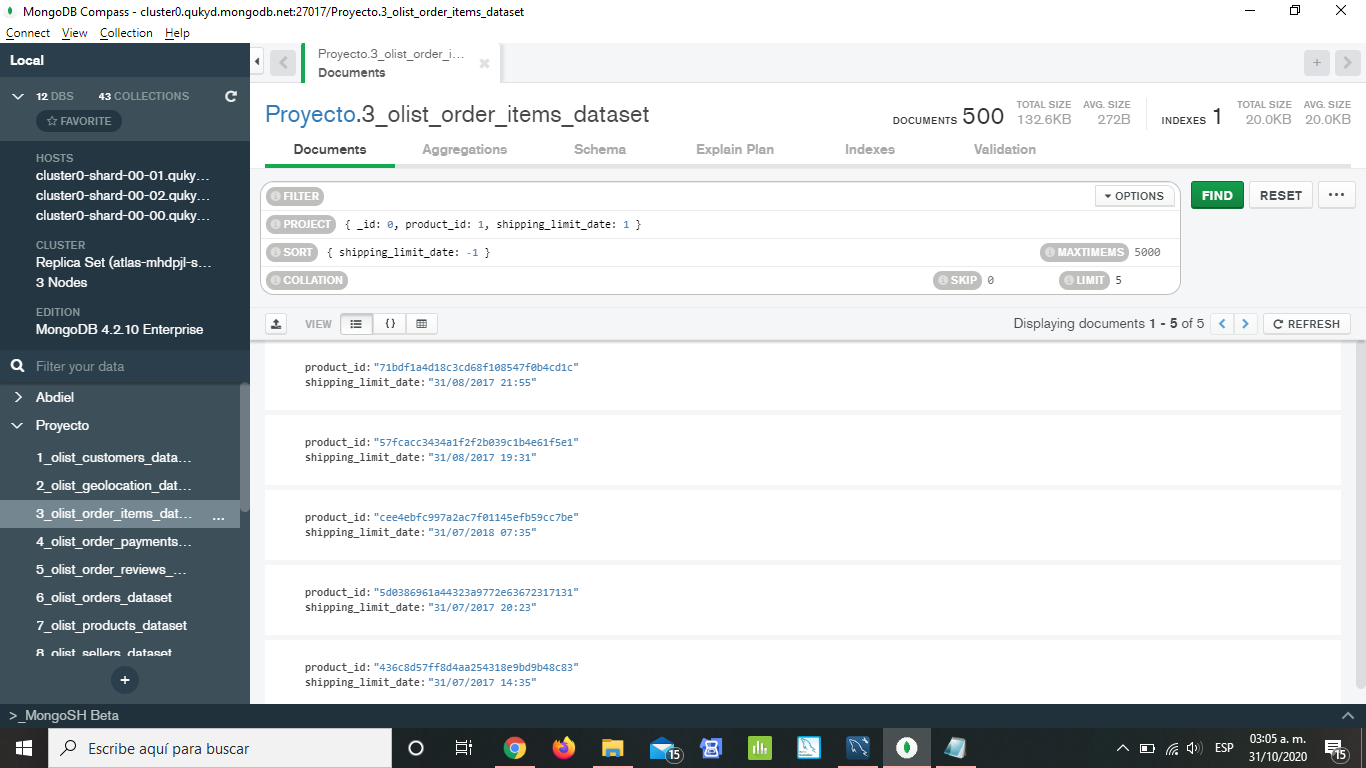
4) LÍMITE CON ORDENAMIENTO. ¿Cuál es el top 5 ítems de orden con la fecha límite de envío más reciente? (3)

MySQL WORKBENCH



SENTENCIA: select product\_id, shipping\_limit\_date from 3\_olist\_order\_items\_dataset order by shipping\_limit\_date desc limit 5;

MONGODB

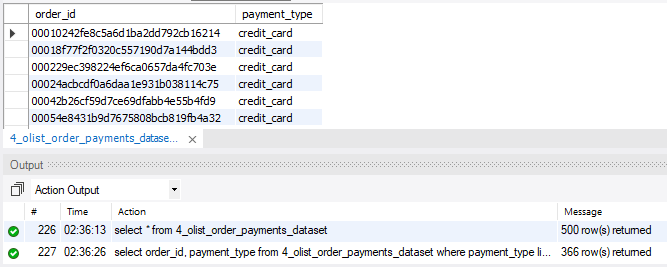


JSON: Project: { \_id: 0, product\_id: 1, shipping\_limit\_date: 1 }

Sort: { shipping\_limit\_date: -1 }

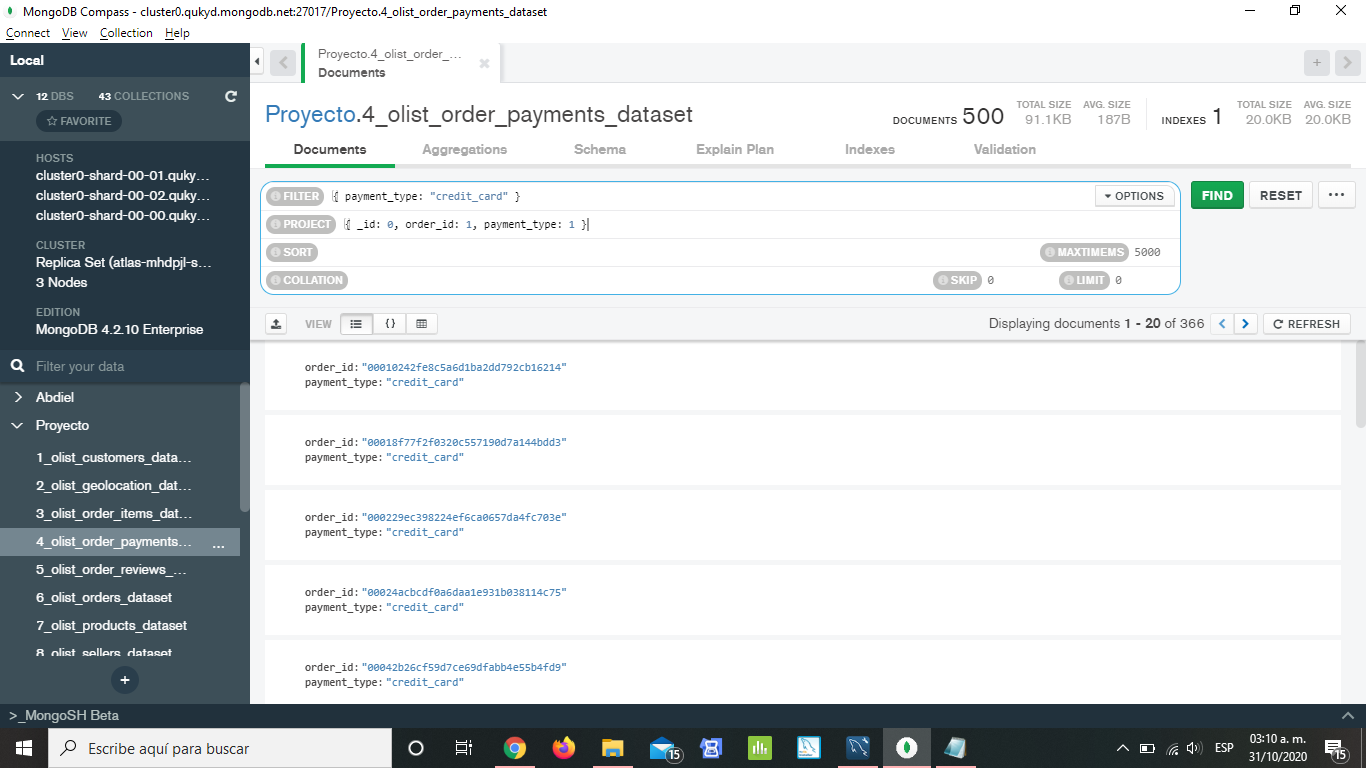
5) PROYECCIÓN CON FILTRO. ¿Cuál es el ID de los pagos cuyo tipo de pago es únicamente por medio de tarjeta de crédito? (4)

MySQL WORKBENCH



SENTENCIA: select order\_id, payment\_type from 4\_olist\_order\_payments\_dataset where payment\_type like 'credit\_card';

MONGODB



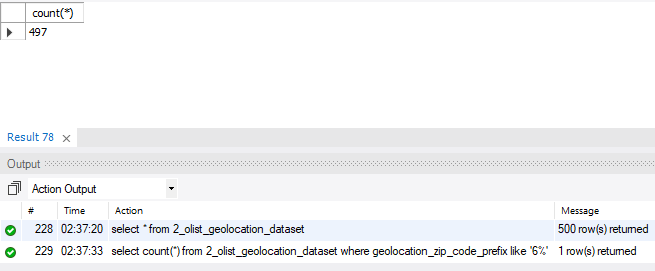
JSON: Filter: { payment\_type: "credit\_card" }

Project: { \_id: 0, order\_id: 1, payment\_type: 1 }

### Agrupamientos y segmentaciones (Post Work II & V)

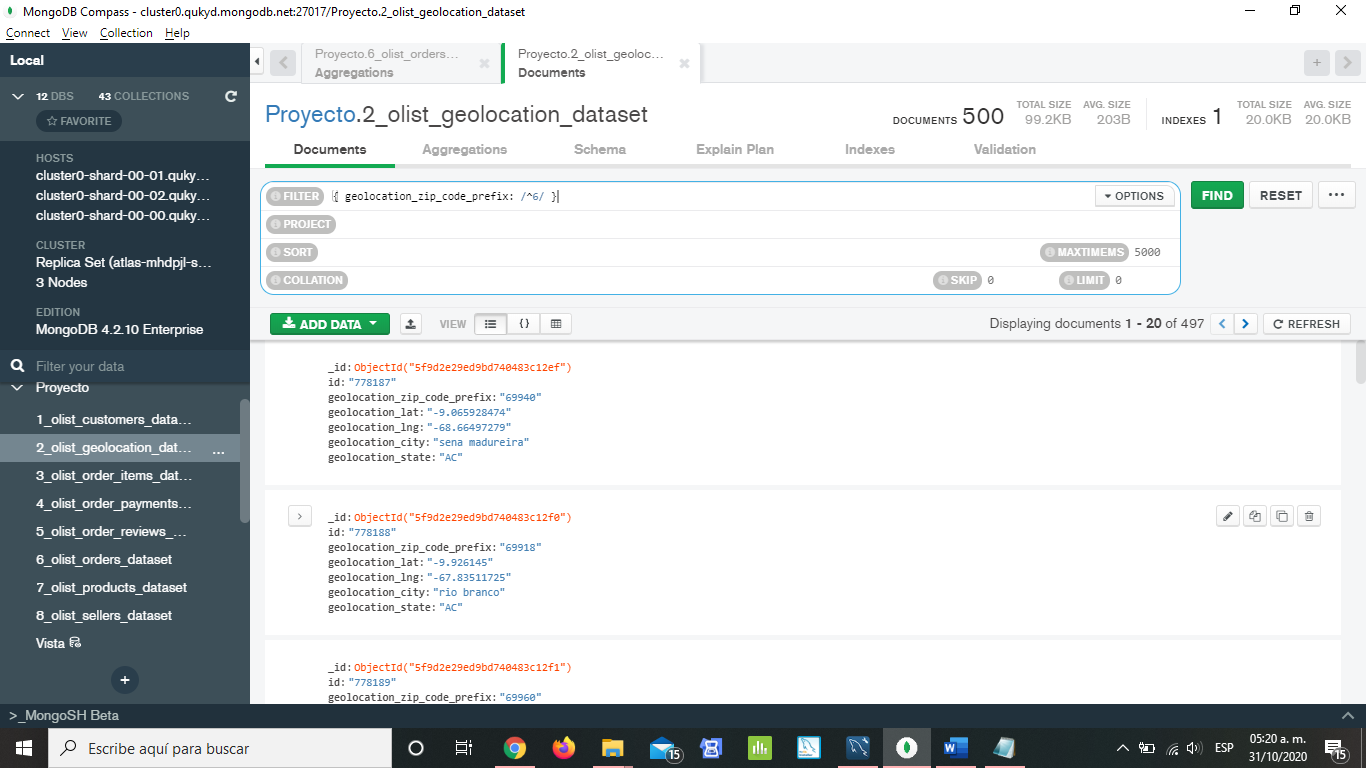
1) PATRONES. ¿Cuántos id hay cuyo prefijo de código postal inicie con el número 6? (2)

MySQL WORKBENCH



SENTENCIA: select count(\*) from 2\_olist\_geolocation\_dataset where geolocation\_zip\_code\_prefix like '6%';

MONGODB



JSON: Filter: { geolocation\_zip\_code\_prefix: /^6/ }

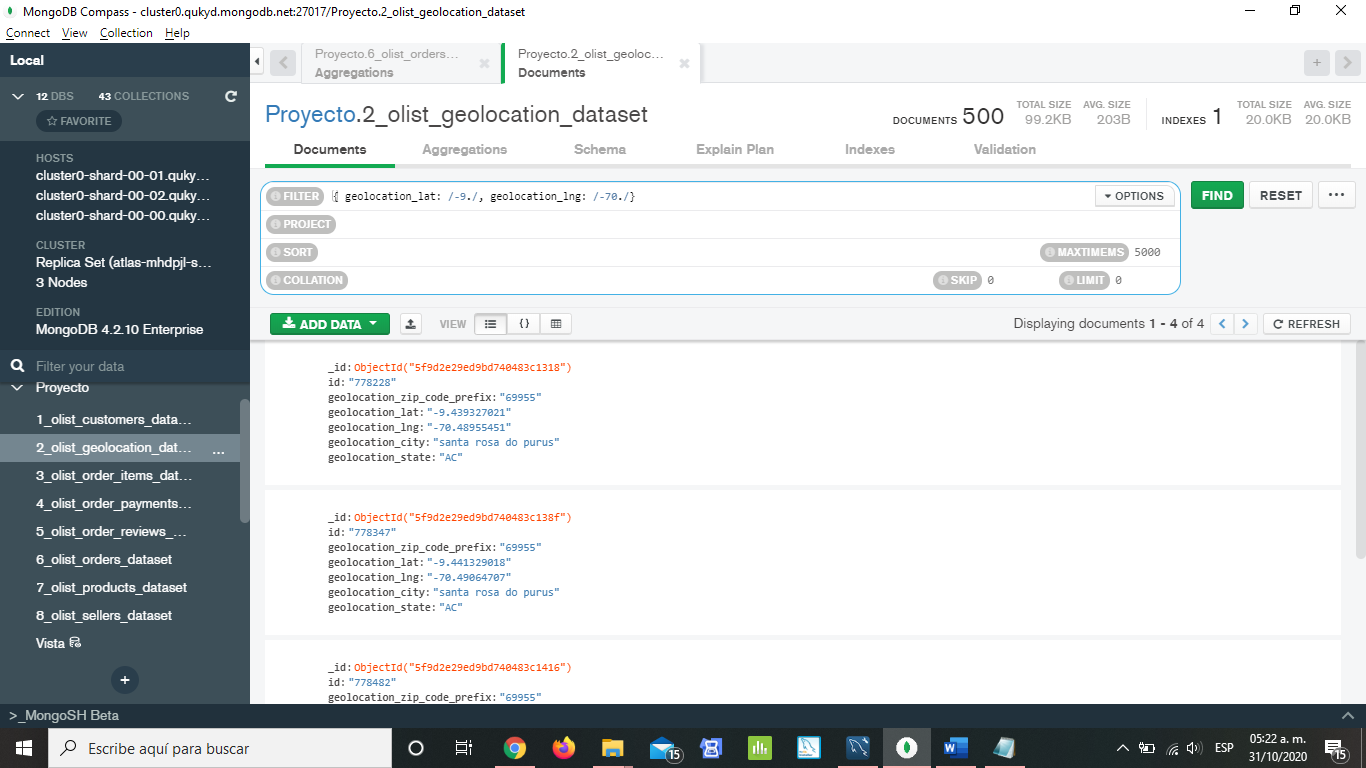
2) PATRONES. ¿Cuál es el total de aquellos id cuya latitud sea -9 y cuya longitud sea -70? (2)

MySQL WORKBENCH



SENTENCIA: select count(\*) from 2\_olist\_geolocation\_dataset where geolocation\_lat like '-9.%' and geolocation\_lng like '-70.%';

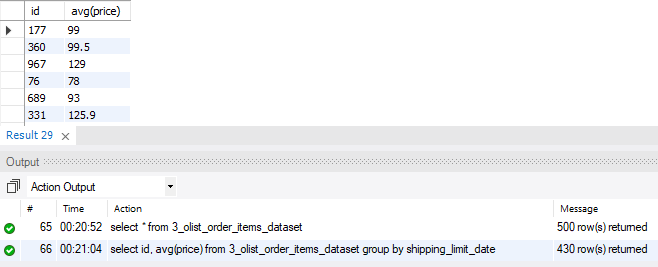
MONGODB



JSON: Filter: { geolocation\_lat: /-9./, geolocation\_lng: /-70./}

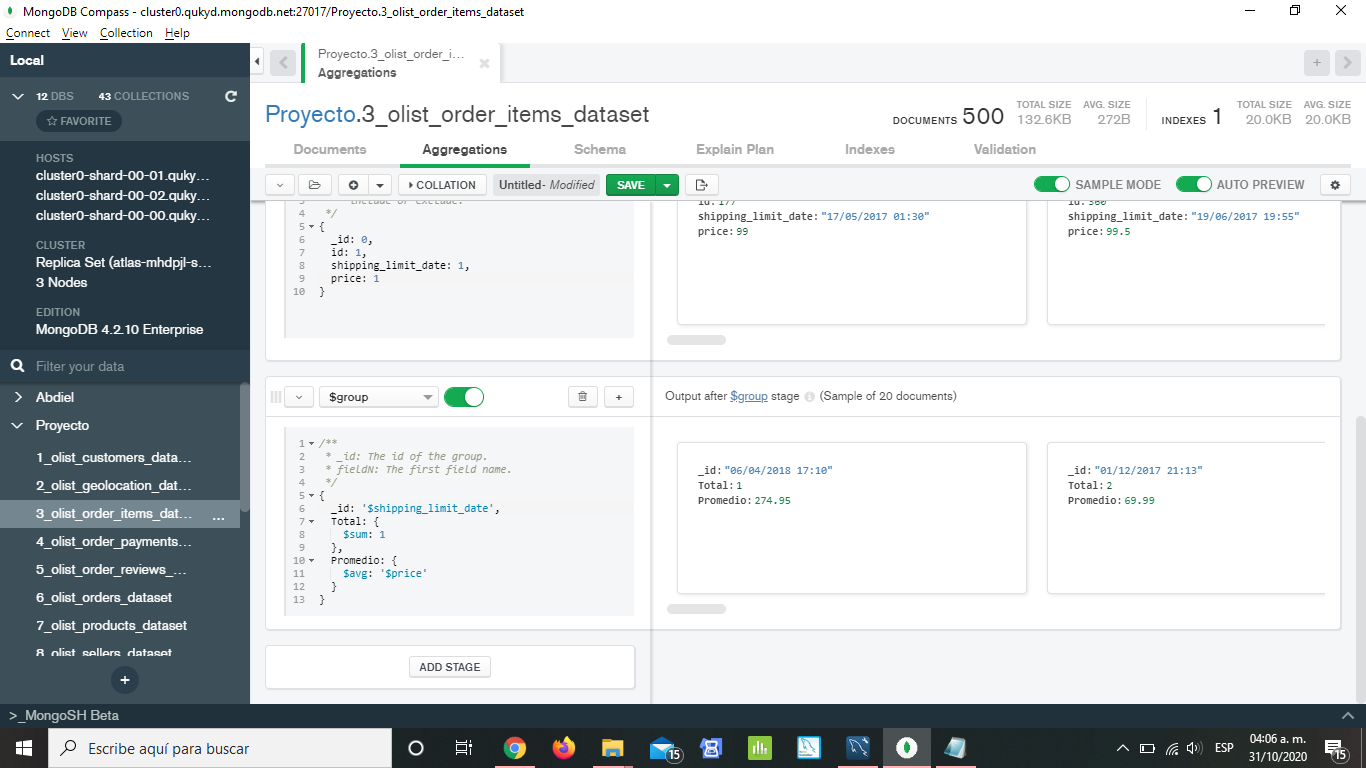
3) AGRUPAMIENTO. ¿Cuál es el precio unitario promedio de los productos agrupados por fecha límite de entrega? (3)

MySQL WORKBENCH



SENTENCIA: select id, avg(price) from 3\_olist\_order\_items\_dataset group by shipping\_limit\_date;

MONGODB



JSON:

[{$project: {

\_id: 0,

id: 1,

shipping\_limit\_date: 1,

price: 1

}}, {$group: {

\_id: '$shipping\_limit\_date',

Total: {

$sum: 1

},

Promedio: {

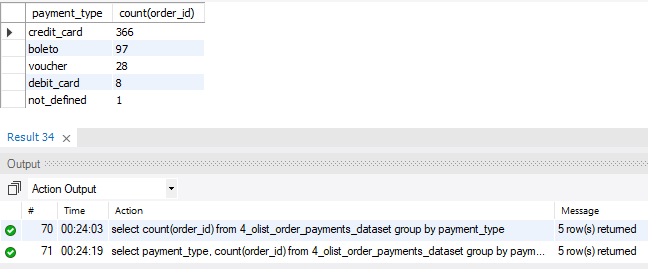
$avg: '$price'

}

}}]

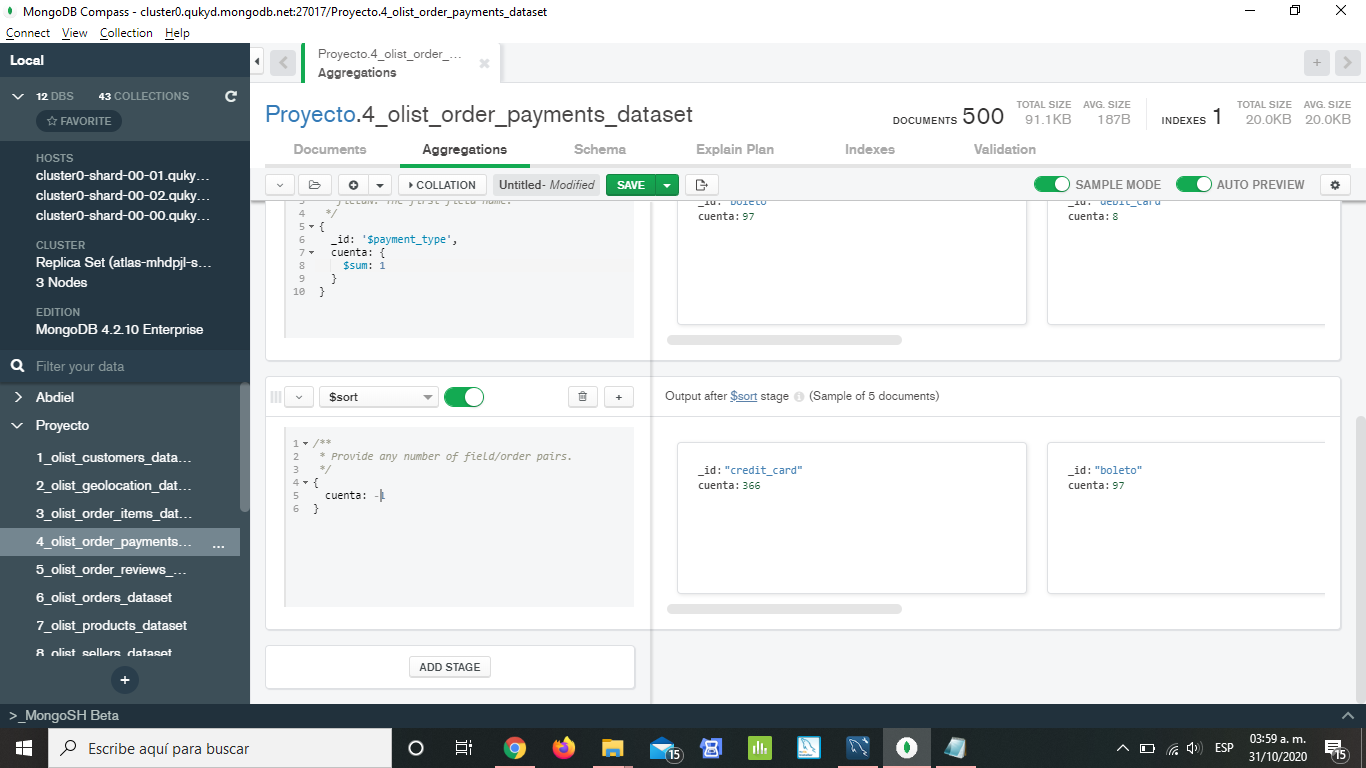
**4) AGRUPAMIENTO. ¿Cuántas órdenes hay por cada tipo de pago? (4)**

MySQL WORKBENCH



SENTENCIA: select payment\_type, count(order\_id) from 4\_olist\_order\_payments\_dataset group by payment\_type;

MONGODB



JSON:

[{$group: {

\_id: '$payment\_type',

cuenta: {

$sum: 1

}

}}, {$sort: {

cuenta: -1

}}]

**5) SUBCONSULTA. ¿Cuántos pagos hay cuyo valor de pago sea mayor al promedio del valor de pago? (4)**

MySQL WORKBENCH



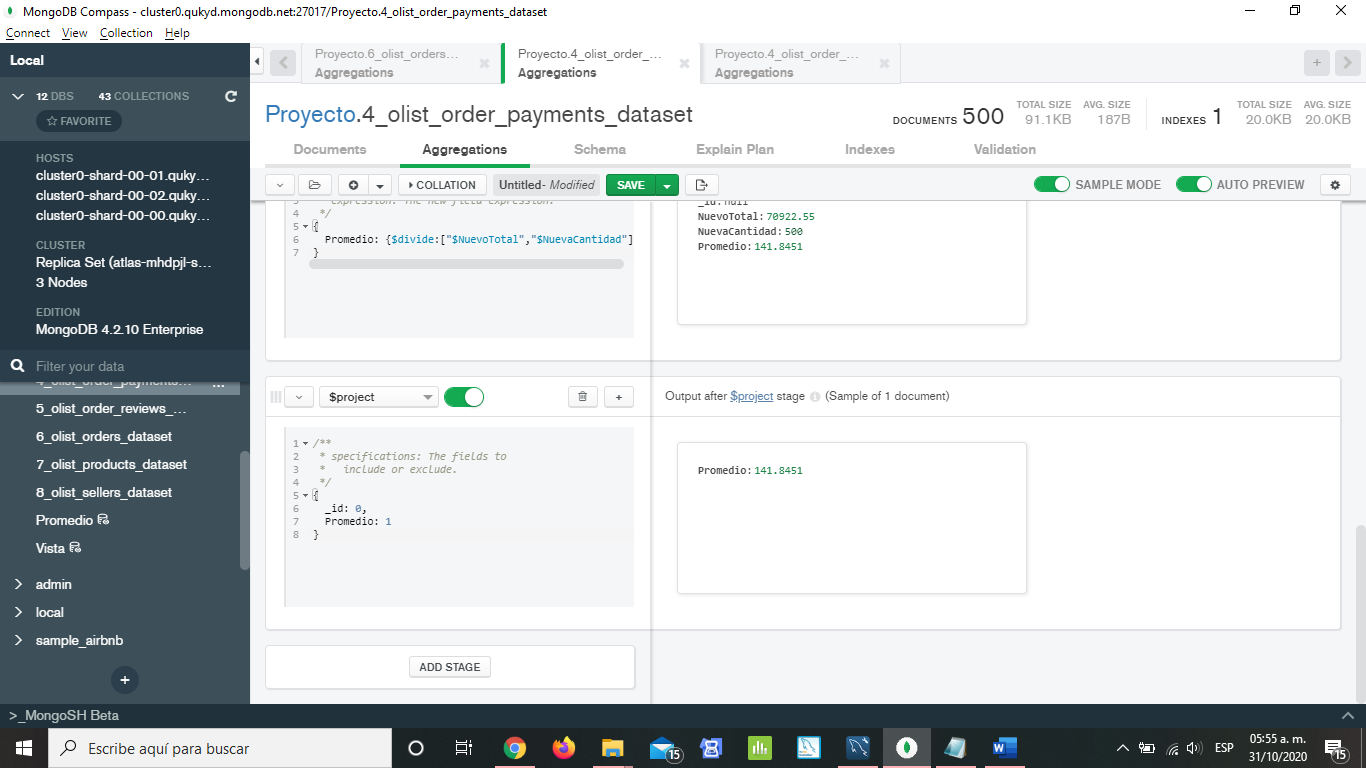
SENTENCIA: select count(payment\_id) as Cantidad

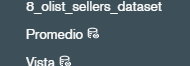
from 4\_olist\_order\_payments\_dataset

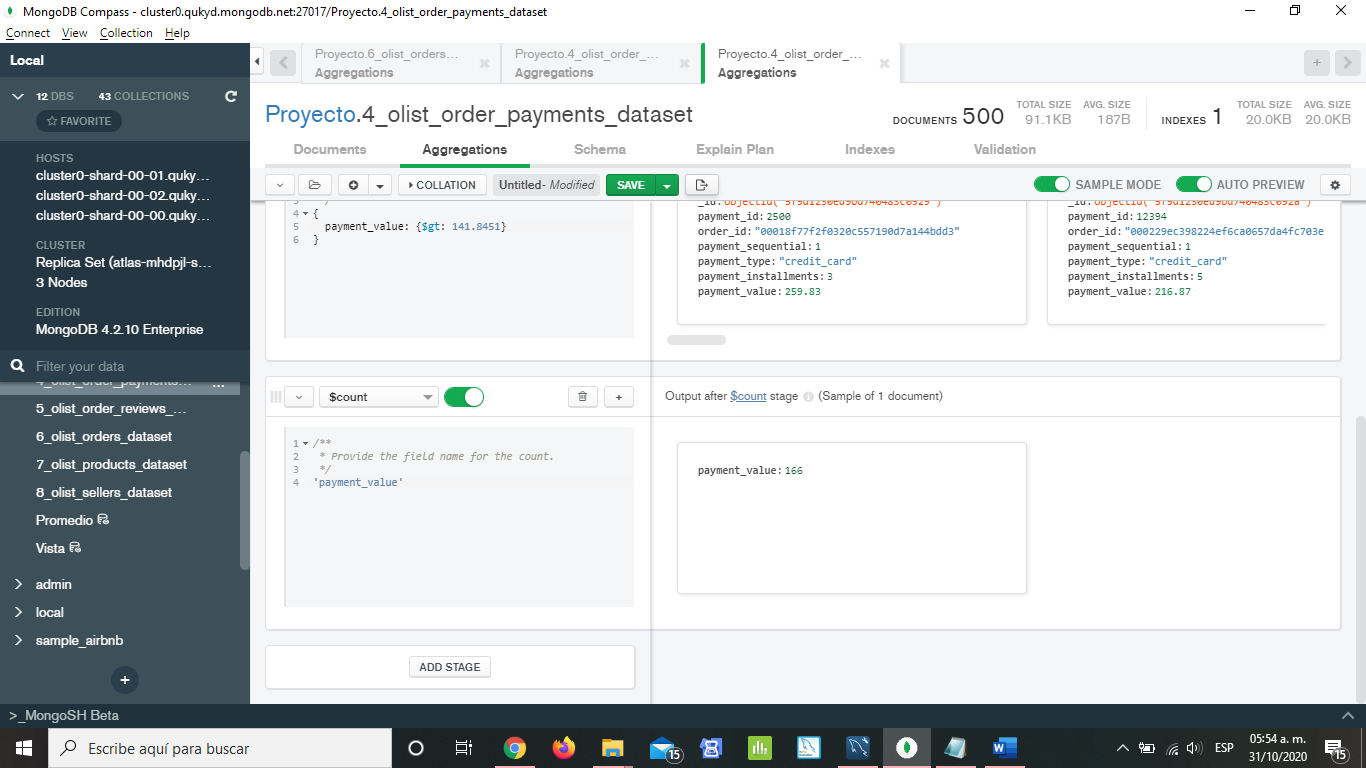
where payment\_value > (select round(avg(payment\_value), 2) as Promedio

from 4\_olist\_order\_payments\_dataset);

MONGODB







JSON: Este se trabajó en dos partes. Primero, se realizó la vista para sacar el promedio y, en base a ese resultado, se realizó la comparación y conteo.

Parte 1.

[{$group: {

\_id: "$payment\_sequential",

Total: {

$sum: "$payment\_value"

},

Cantidad: {

$sum: 1

}

}}, {$group: {

\_id: null,

NuevoTotal: {

$sum: "$Total"

},

NuevaCantidad: {

$sum: "$Cantidad"

}

}}, {$addFields: {

Promedio: {$divide:["$NuevoTotal","$NuevaCantidad"]}

}}, {$project: {

\_id: 0,

Promedio: 1

}}]

Parte 2.

[{$match: {

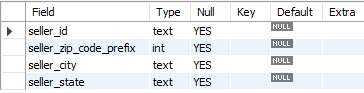
payment\_value: {$gt: 141.8451}

}}, {$count: 'payment\_value'}]

### Relaciones entre tablas (Post Work III & VI)

1) DESCRIPCIÓN DE TABLAS. Describe la tabla de sellers\_dataset. (8)

MySQL WORKBENCH

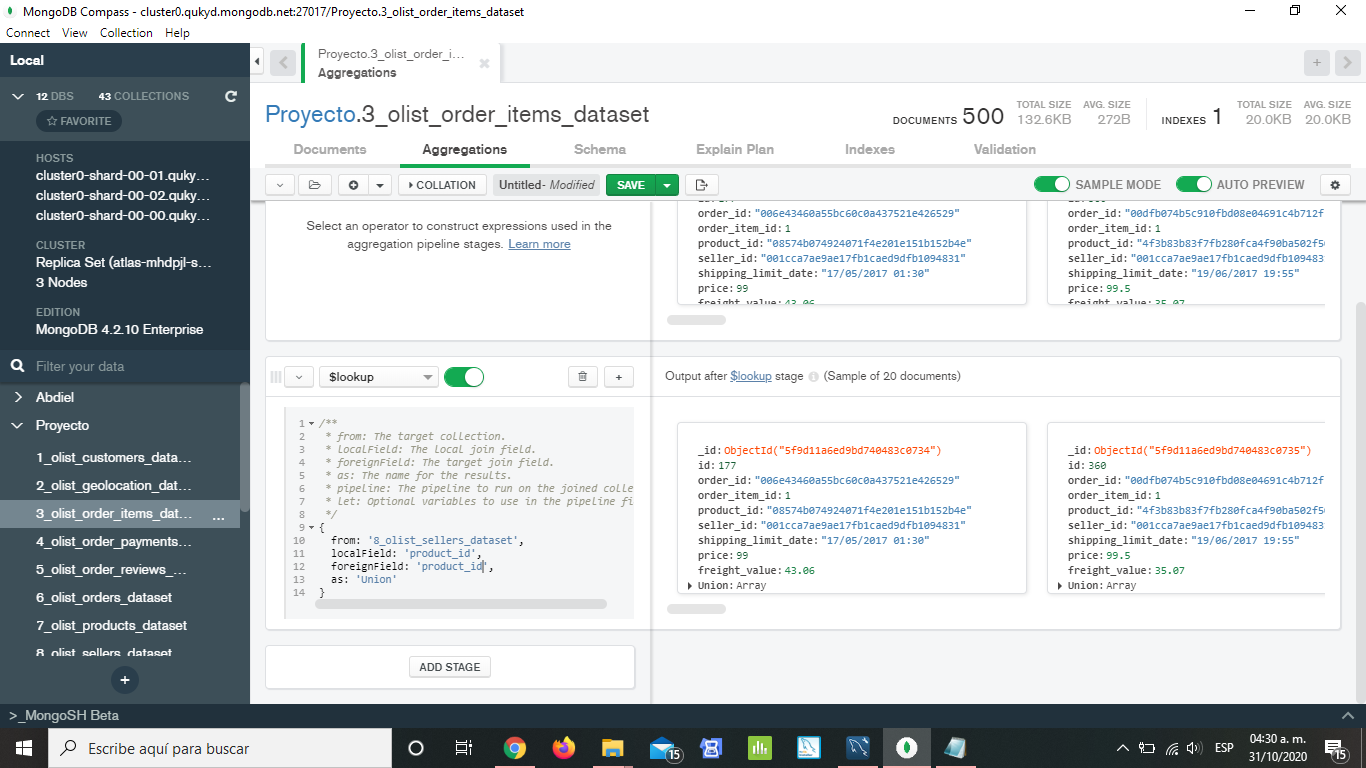


SENTENCIA: describe 8\_olist\_sellers\_dataset;

MONGODB

1) Une las tablas order\_items y sellers\_dataset (3 con 8)

JSON: Proyección por medio de agregación ($lookup)



JSON:

[{$lookup: {

from: '8\_olist\_sellers\_dataset',

localField: 'product\_id',

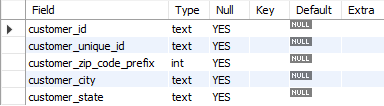
foreignField: 'product\_id',

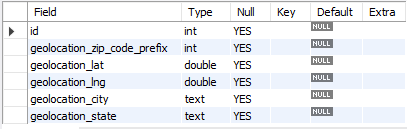
as: 'Union'

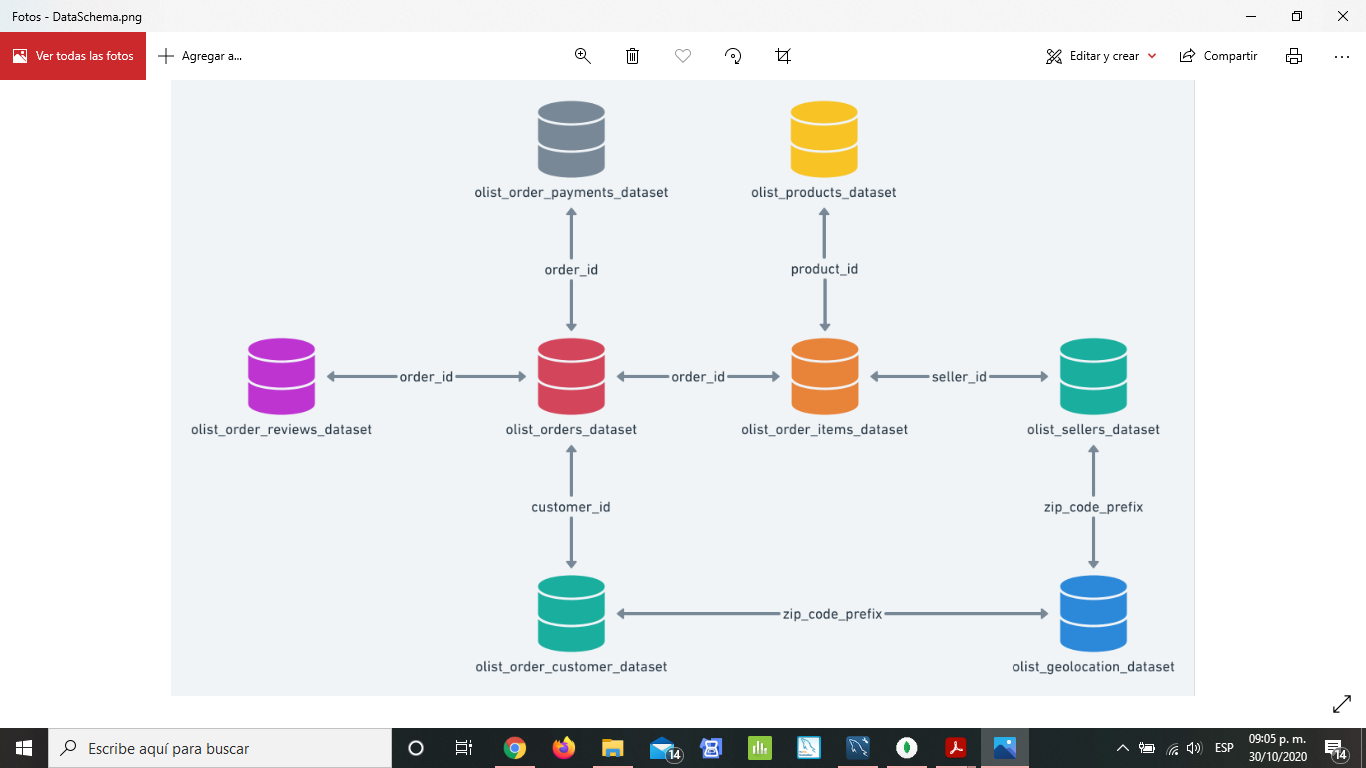
}}]

2) TIPOS DE DATOS Y RELACIONES. Describe las tablas customers\_dataset y geolocation\_dataset y, ¿qué relación hay entre ellas? (1 con 2)

MySQL WORKBENCH

**





SENTENCIA: describe 1\_olist\_customers\_dataset;

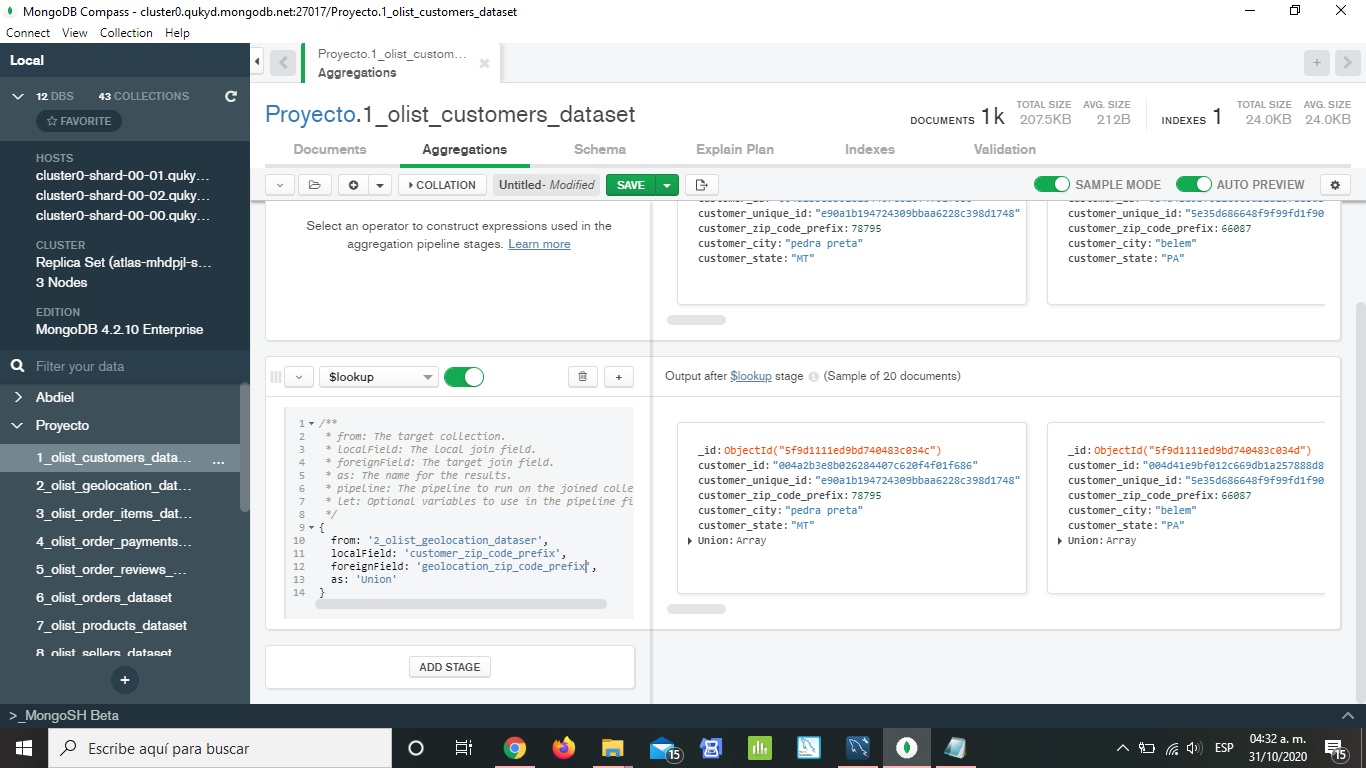
describe 2\_olist\_geolocation\_dataset;

Relación 1:1.

MONGODB

2) Une las tablas customers\_dataset y geolocation\_dataset (1 con 2)

JSON: Proyección por medio de agregación ($lookup)



JSON:

[{$lookup: {

from: '2\_olist\_geolocation\_dataser',

localField: 'customer\_zip\_code\_prefix',

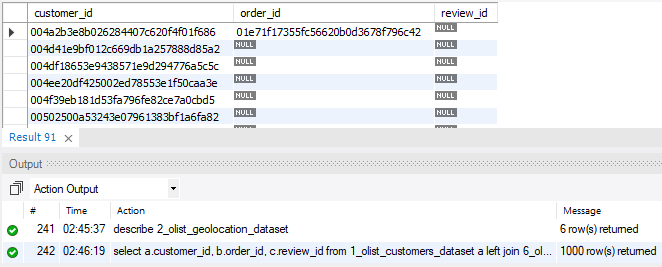
foreignField: 'geolocation\_zip\_code\_prefix',

as: 'Union'

}}]

3) JOIN. ¿Cuál es la reseña de cada cliente? (1, 6, 5)

MySQL WORKBENCH



SENTENCIA: select a.customer\_id, b.order\_id, c.review\_id from 1\_olist\_customers\_dataset a

left join 6\_olist\_orders\_dataset b

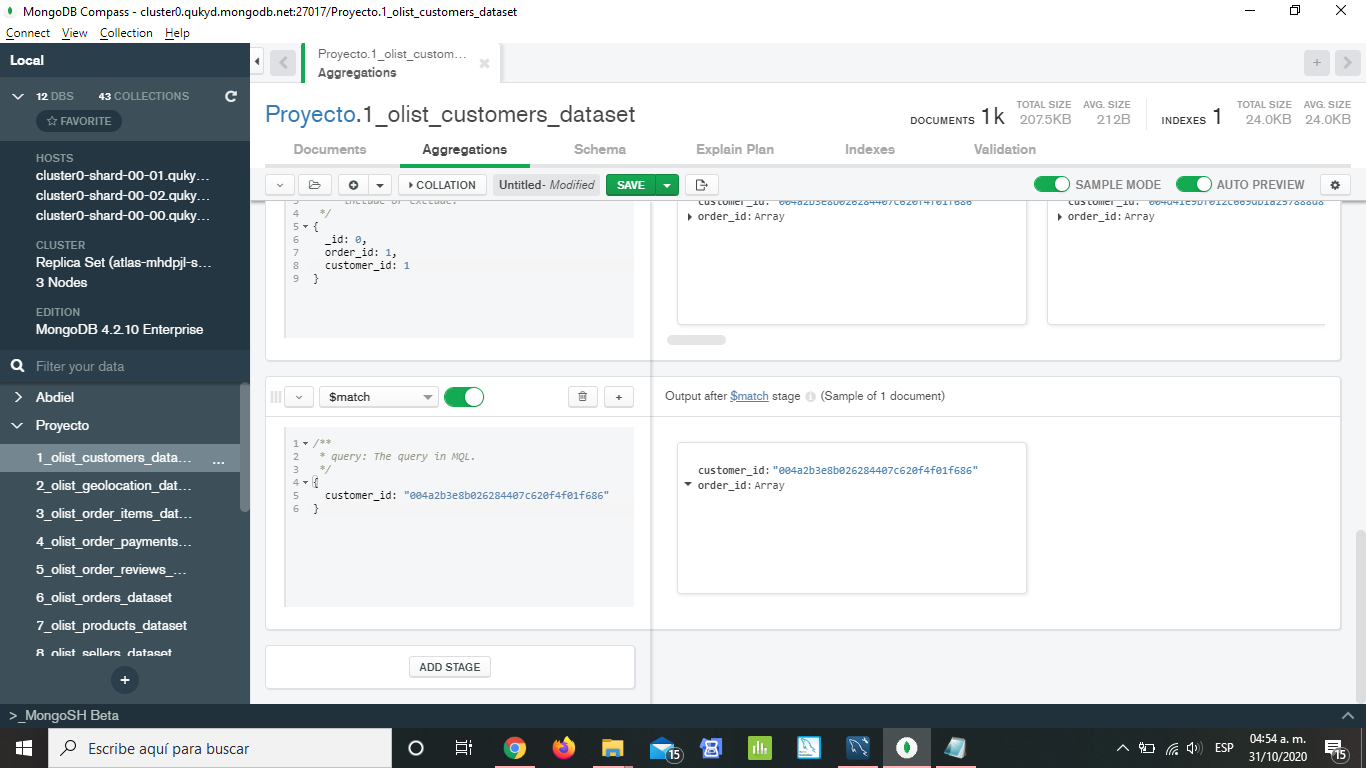
on a.customer\_id = b.customer\_id

left join 5\_olist\_order\_reviews\_dataset c

on b.order\_id = c.order\_id;

Por lo tanto, debido a la reducción de clientes, se encontró que los clientes de la base de datos no hicieron comentario alguno y sus órdenes no están dentro de la tabla homónima.

MONGODB



JSON:

[{$lookup: {

from: '6\_olist\_orders\_dataset',

localField: 'customer\_id',

foreignField: 'customer:id',

as: 'order\_id'

}}, {$lookup: {

from: '5\_olist\_order\_reviews\_dataset',

localField: 'order\_id',

foreignField: 'order\_id',

as: 'review\_id'

}}, {$project: {

\_id: 0,

order\_id: 1,

customer\_id: 1

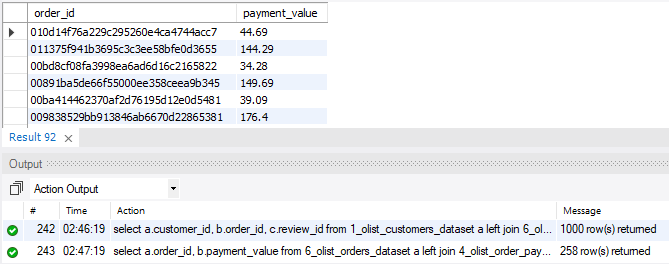
}}, {$match: {

customer\_id: "004a2b3e8b026284407c620f4f01f686"

}}]

**4) JOIN. ¿Cuál es el valor de pago de cada orden? (4 y 6)**

MySQL WORKBENCH



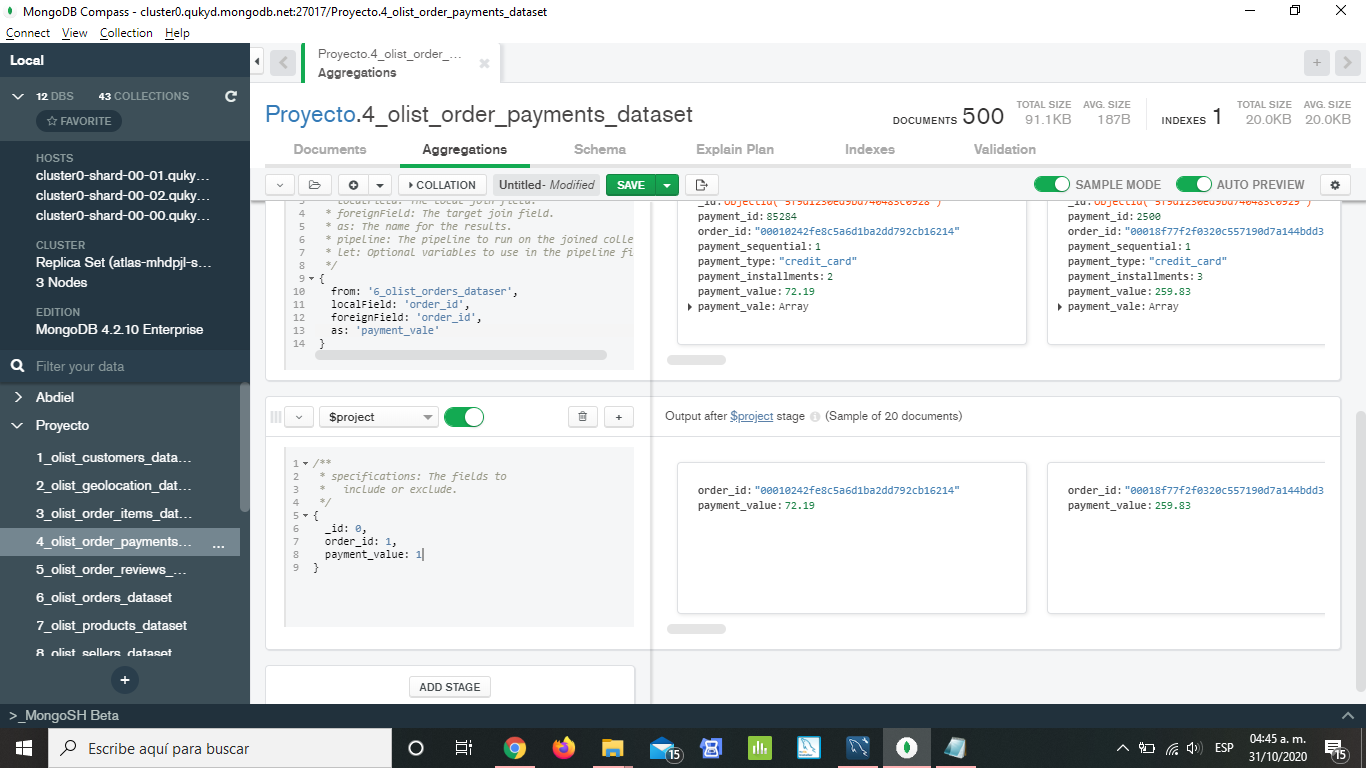
SENTENCIA: select a.order\_id, b.payment\_value from 6\_olist\_orders\_dataset a

left join 4\_olist\_order\_payments\_dataset b

on a.order\_id = b.order\_id

where payment\_value != 'Null';

MONGODB



JSON:

[{$lookup: {

from: '6\_olist\_orders\_dataser',

localField: 'order\_id',

foreignField: 'order\_id',

as: 'payment\_vale'

}}, {$project: {

\_id: 0,

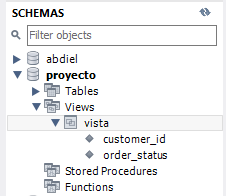
order\_id: 1,

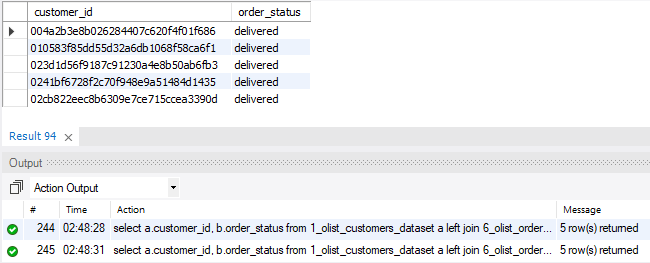
payment\_value: 1

}}]

5) VISTA. Haz una vista que conteste a la pregunta: ¿Cuál es el estatus de la orden de cada cliente excepto los que sean nulos? (1 y 6)

MySQL WORKBENCH





SENTENCIA: create view vista as

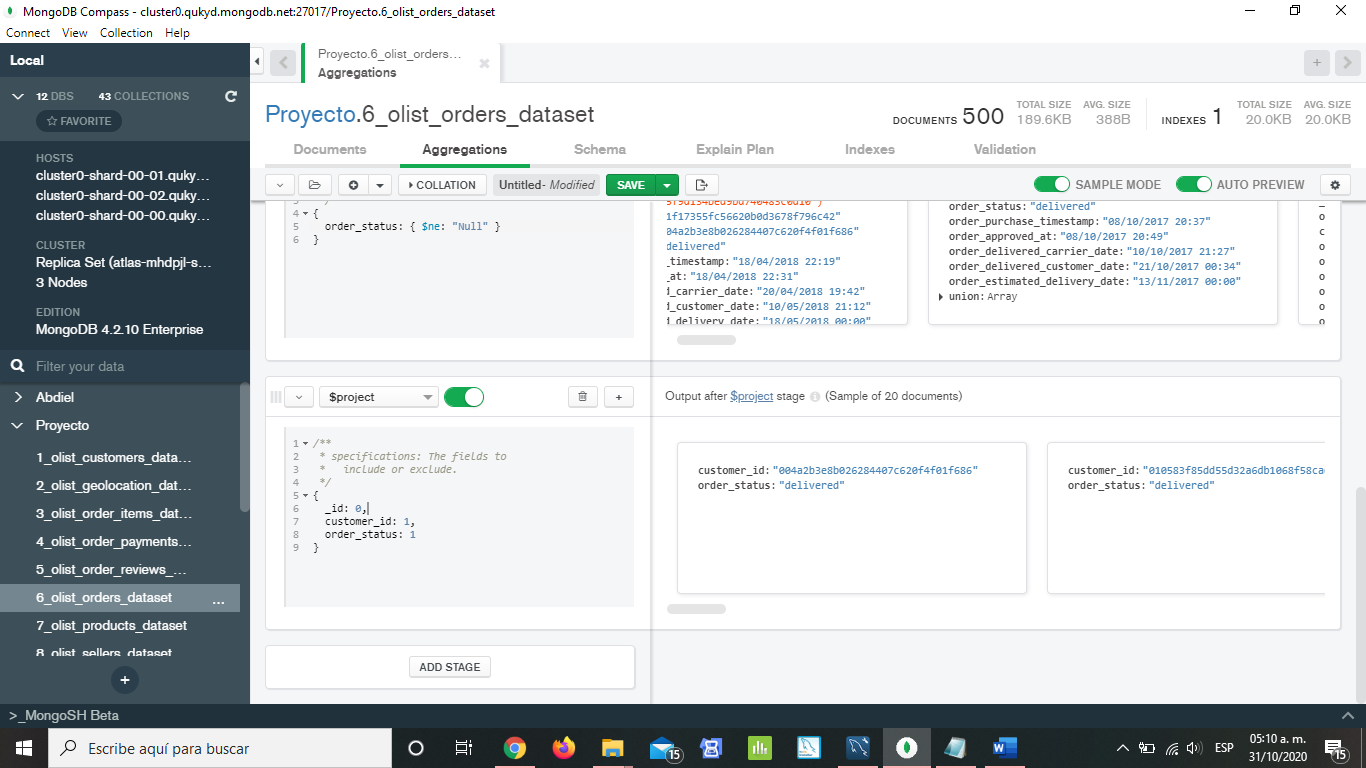
(select a.customer\_id, b.order\_status from 1\_olist\_customers\_dataset a

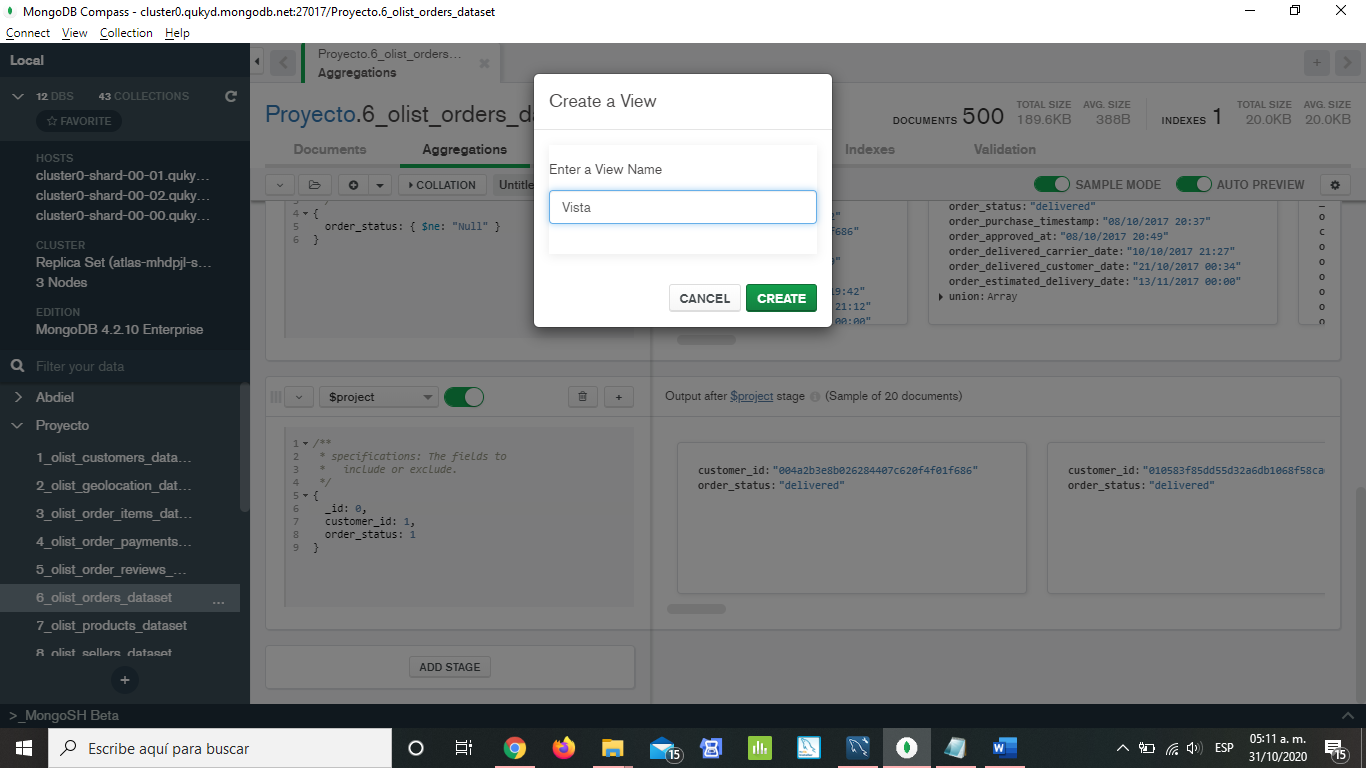
left join 6\_olist\_orders\_dataset b

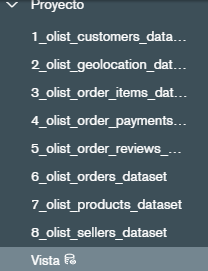
on a.customer\_id = b.customer\_id

where order\_status != 'Null');

MONGODB







JSON:

[{$lookup: {

from: '1\_olist\_customers\_dataser',

localField: 'customer\_id',

foreignField: 'customer\_id',

as: 'union'

}}, {$match: {

order\_status: { $ne: "Null" }

}}, {$project: {

\_id: 0,

customer\_id: 1,

order\_status: 1

}}]

### Resolución del problema

* ¿Qué clientes ya han sido satisfechos al haber registrado su estatus como entregado?

Un total de 487 de 500. Si bien, esta es una inmensa mayoría, se debe dar monitoreo, tomando en cuenta la fecha de evaluación con respecto a las fechas en que se debió haber hecho llegar el producto al cliente.

* ¿Cuál es el método de pago preferido de los usuarios?

Definitivamente la tarjeta de crédito, ya que cuenta con 269 usuarios más que el método que va en segundo lugar, que es por medio de boleto.

* ¿Cuáles son los pagos que están por encima del promedio de su propio valor?

De los 500 registros, se obtiene que 166 está por encima del promedio.

* ¿Cuál es el valor de pago de cada orden?

Varía dependiendo cada orden, sin embargo, esta se puede visualizar bien tanto en MySQL Workbench como en MongoDB, ya que allí pueden visualizarse los 258 registros/documentos.

# Conclusión

El conjunto de preguntas iniciales, en efecto, se responden con la solución de una serie ejercicios puntuales que se formularon a lo largo de los Post Works. Gracias a los Sistemas Gestores de Bases de Datos - en mi caso, reconozco a MySQL ya que fue el que más se me facilitó - podemos gestionar cantidades abrumadoras de información hasta obtener una interpretación de la realidad. En efecto, siempre va a haber sesgo y la perspectiva de todas las personas que interactúan con las bases de datos, ya sean clientes cuya información formará parte de un registro o el propio analista de datos, siempre influye en la toma de decisiones. Es, pues, necesario apegarse al potencial de estas herramientas de gestión y, en especial, aprenden a abstraer el pensamiento de tal manera que se uno pueda comunicarse directamente con todo un sistema.

# Crédito a la Base de Datos

## Base de Datos Original

* kaggle. (s.f.). *Brazilian E-Commerce Public Dataset by Olist*. Recuperado el 27 de Octubre de 2020, de <https://www.kaggle.com/olistbr/brazilian-ecommerce>

## Base de Datos Utilizada

La base de datos utilizada para la realización de este proyecto se redujo de a 1000 registros/documentos para la primer tabla y a 500 para las demás, esto con el fin de facilitar el proceso de carga de datos a la base de datos local.

**La base de datos utilizada en la realización de este proyecto se envió directo al hilo de Slack de Abd CJaén.**