

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	3
Objetivo del curso	3
Objetivo del proyecto	3
Contexto	3
ERR (Diagrama Entidad - Relación)	4
Carga de Bases de Datos	4
MySQL Workbench	4
MongoDB Compass	5
Solución	5
Plan de Solución	5
Procedimientos	6
Exploración y consultas iniciales (Post Work I & IV)	6
Agrupamientos y segmentaciones (Post Work II & V)	10
Relaciones entre tablas (Post Work III & VI)	16
Resolución del problema	24
Conclusión	24
Crédito a la Base de Datos	24
Base de Datos Original	24
Base de Datos Utilizada	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

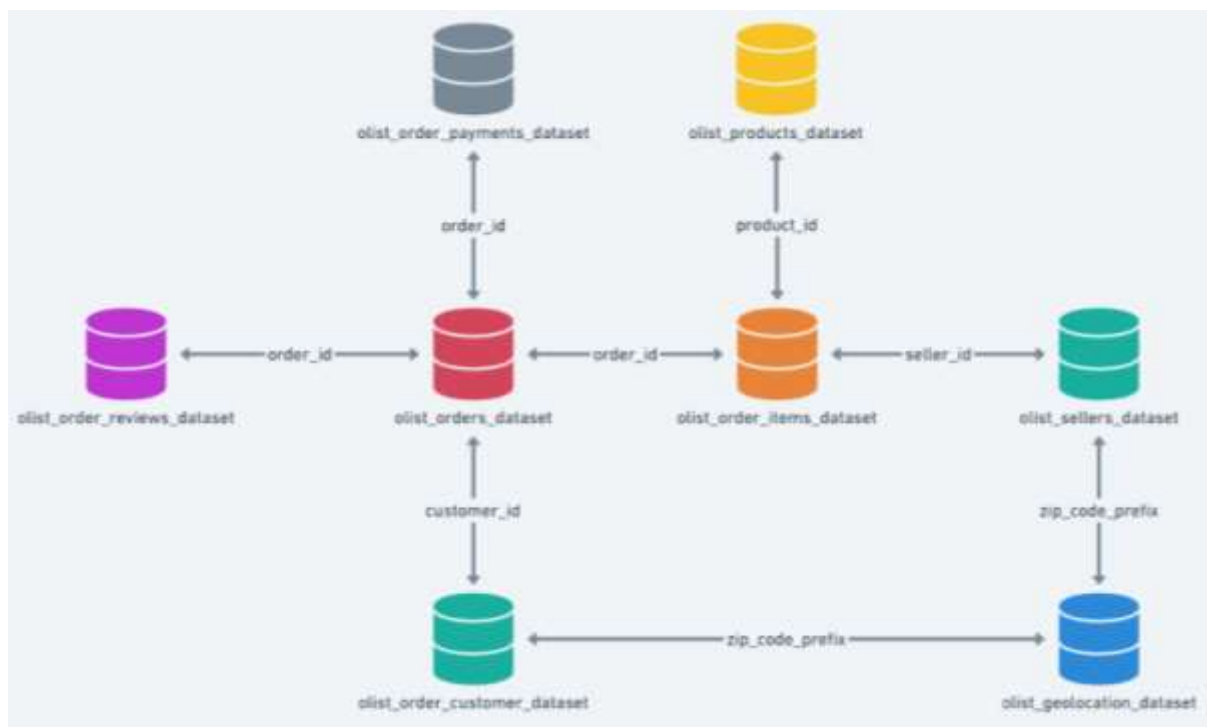
El presente trabajo fue realizado en base a una base de datos de una empresa brasileña de ecommerce, publicada por cortesía de Olist. Sus características permiten la visualización de su información a través de múltiples dimensiones: desde el estatus de la orden, el precio, el pago y el rendimiento del 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.

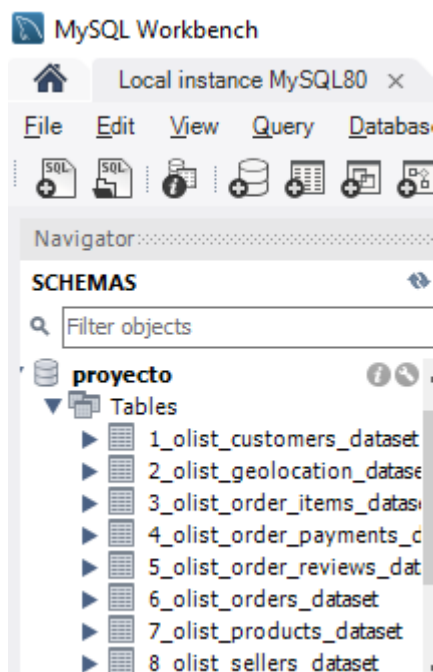
La base de datos, que está pensada para consultarse de modo relacional (SQL), será, además, tratada de manera no relacional (No SQL).

ERR (Diagrama Entidad - Relación)

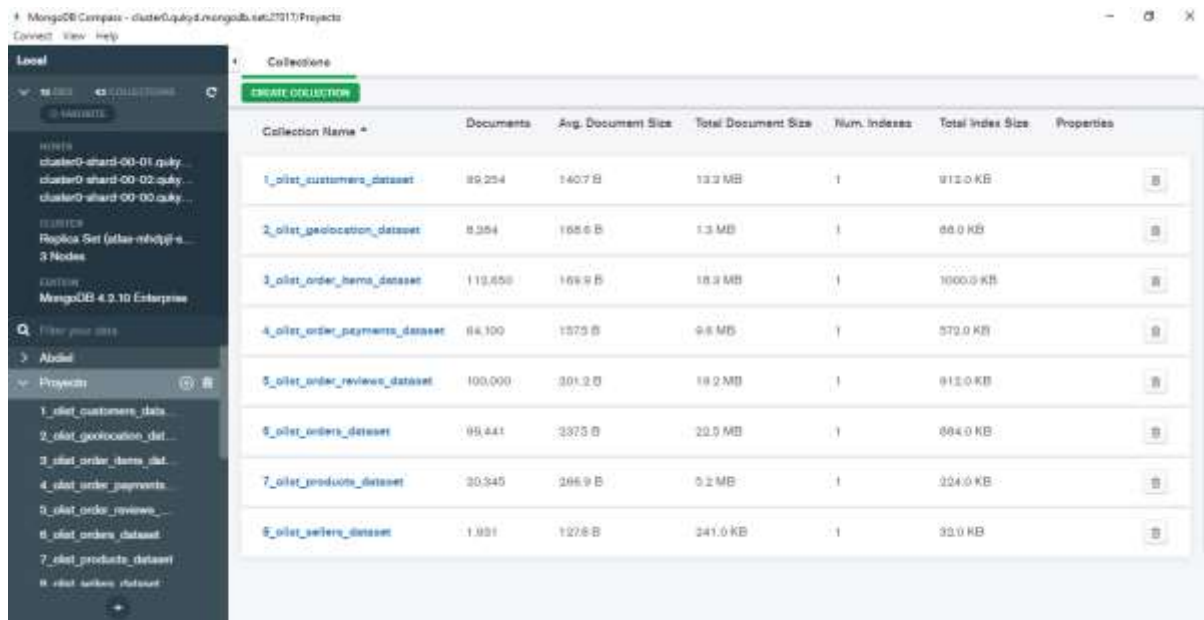


Carga de Bases de Datos

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

	customer_id	order_status
▶	004a2b3e8b026284407c620f4f01f686	delivered
	010583f85dd55d32a6db1068f58ca6f1	delivered
	023d1d56f9187c91230a4e8b50ab6fb3	delivered
	0241bf6728f2c70f948e9a51484d1435	delivered
	02cb822eec8b6309e7ce715ccea3390d	delivered
	02d2c49ef9837bf18d71940dd7cf97f	delivered

#	Time	Action	Message
✓ 217	02:31:08	select * from 6_olist_orders_dataset	500 row(s) returned
✓ 218	02:31:46	select customer_id, order_status from 6_olist_orders_dataset	500 row(s) returned

SENTENCIA: select customer_id, order_status from 6_olist_orders_dataset;

MONGODB

customer_id: "004a2b3e8b026284407c620f4f01f686"	order_status: "delivered"
customer_id: "010583f85dd55d32a6db1068f58ca6f1"	order_status: "delivered"
customer_id: "023d1d56f9187c91230a4e8b50ab6fb3"	order_status: "delivered"
customer_id: "0241bf6728f2c70f948e9a51484d1435"	order_status: "delivered"
customer_id: "02cb822eec8b6309e7ce715ccea3390d"	order_status: "delivered"
customer_id: "02d2c49ef9837bf18d71940dd7cf97f"	order_status: "delivered"

JSON: Project: { _id: 0, customer_id: 1, order_status: 1 }

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

MySQL WORKBENCH

	customer_id
▶	004a2b3e8b026284407c620f4f01f686
	010583f85dd55d32a6db1068f58ca6f1
	023d1d56f9187c91230a4e8b50ab6fb3
	0241bf6728f2c70f948e9a51484d1435
	02cb822eec8b6309e7ce715ccea3390d
	02d2c49ef9837fbf18d71940dd7cf97f

6_olist_orders_dataset 70 x			
Output			
Action Output			
#	Time	Action	Message
✓ 218	02:31:46	select customer_id, order_status from 6_olist_orders_dataset	500 row(s) returned
✓ 219	02:32:24	select customer_id from 6_olist_orders_dataset where order_status like 'delivered'	487 row(s) returned

SENTENCIA: select customer_id from 6_olist_orders_dataset where order_status like 'delivered';

MONGODB

FILTER { order_status: 'delivered' } PROJECT { _id: 1, customer_id: 1, order_status: 1 } SORT COLLATION MAXTIME MS: 1000 SKIP 0 LIMIT 0	FIND RESET Displaying documents 1 - 20 of 487 REFRESH
customer_id: "004a2b3e8b026284407c620f4f01f686" order_status: "delivered"	
customer_id: "010583f85dd55d32a6db1068f58ca6f1" order_status: "delivered"	
customer_id: "023d1d56f9187c91230a4e8b50ab6fb3" order_status: "delivered"	
customer_id: "0241bf6728f2c70f948e9a51484d1435" order_status: "delivered"	
customer_id: "02cb822eec8b6309e7ce715ccea3390d" order_status: "delivered"	

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

	id	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
▶	778436	69945	-10.08000105	-67.05371944	acrelandia	AC
	778600	69945	-9.937884	-66.928913	acrelandia	AC
	778427	69945	-10.07466858	-67.05516175	acrelandia	AC
	778390	69935	-10.94051041	-69.55865329	assis brasil	AC
	778598	69935	-10.94140648	-69.56741561	assis brasil	AC
	778506	69935	-10.94128913	-69.56740734	assis brasil	AC

2_olist_geolocation_dataset 72 x

Output

Action Output

#	Time	Action	Message
✓ 222	02:34:02	select * from 2_olist_geolocation_dataset	500 row(s) returned
✓ 223	02:34:20	select * from 2_olist_geolocation_dataset order by geolocation_city	500 row(s) returned

SENTENCIA: select * from 2_olist_geolocation_dataset order by geolocation_city;
MONGODB

The screenshot shows the MongoDB Compass interface. At the top, there's a query editor with the following query: `{ geolocation_city: 1 }`. Below the query editor, there are buttons for 'ADD DATA', 'VIEW', and 'FILTER'. The results section shows three documents displayed in a JSON format. The first document is for 'acrelandia' with ID '778427'. The second document is for 'acrelandia' with ID '778436'. The third document is for 'assis brasil' with ID '778598'.

```

{ "_id": ObjectId("5f9d18aed9bd740463a79e"),
  "id": 778427,
  "geolocation_zip_code_prefix": 69945,
  "geolocation_lat": -10.07466858,
  "geolocation_lng": -67.05516175,
  "geolocation_city": "acrelandia",
  "geolocation_state": "AC" }

{ "_id": ObjectId("5f9d18aed9bd740463a79e"),
  "id": 778436,
  "geolocation_zip_code_prefix": 69945,
  "geolocation_lat": -10.08000105,
  "geolocation_lng": -67.05371944,
  "geolocation_city": "acrelandia",
  "geolocation_state": "AC" }

{ "_id": ObjectId("5f9d18aed9bd740463a79e"),
  "id": 778598,
  "geolocation_zip_code_prefix": 69935,
  "geolocation_lat": -10.94140648,
  "geolocation_lng": -69.56741561,
  "geolocation_city": "assis brasil",
  "geolocation_state": "AC" }

```

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

	product_id	shipping_limit_date
▶	71bdf1a4d18c3cd68f108547f0b4cd1c	31/08/2017 21:55
	57fcacc3434a1f2f2b039c1b4e61f5e1	31/08/2017 19:31
	cee4ebfc997a2ac7f01145efb59cc7be	31/07/2018 07:35
	5d0386961a44323a9772e63672317131	31/07/2017 20:23
	436c8d57ff8d4aa254318e9bd9b48c83	31/07/2017 14:35

3_olist_order_items_dataset 97 x

Output

Action Output

#	Time	Action	Message
✓ 248	03:07:02	select product_id, shipping_limit_date from 3_olist_order_items_dataset where shipping_lim...	5 row(s) returned
✓ 249	03:07:18	select product_id, shipping_limit_date from 3_olist_order_items_dataset order by shipping_li...	5 row(s) returned

SENTENCIA: select product_id, shipping_limit_date from
3_olist_order_items_dataset order by shipping_limit_date desc limit 5;

MONGODB

The screenshot shows the MongoDB Compass interface. The query editor displays the following query:

```
{ "_id": 0, "product_id": 1, "shipping_limit_date": 1 }
```

The sort criteria is set to:

```
{ "shipping_limit_date": -1 }
```

The results tab shows 5 documents, sorted by shipping_limit_date in descending order. The documents are:

```
{
  "product_id": "71bdf1a4d18c3cd68f108547f0b4cd1c",
  "shipping_limit_date": "31/08/2017 21:55"
}
```

```
{
  "product_id": "57fcacc3434a1f2f2b039c1b4e61f5e1",
  "shipping_limit_date": "31/08/2017 19:31"
}
```

```
{
  "product_id": "cee4ebfc997a2ac7f01145efb59cc7be",
  "shipping_limit_date": "31/07/2018 07:35"
}
```

```
{
  "product_id": "5d0386961a44323a9772e63672317131",
  "shipping_limit_date": "31/07/2017 20:23"
}
```

```
{
  "product_id": "436c8d57ff8d4aa254318e9bd9b48c83",
  "shipping_limit_date": "31/07/2017 14:35"
}
```

JSON: Project: { _id: 0, product_id: 1, shipping_limit_date: 1 }

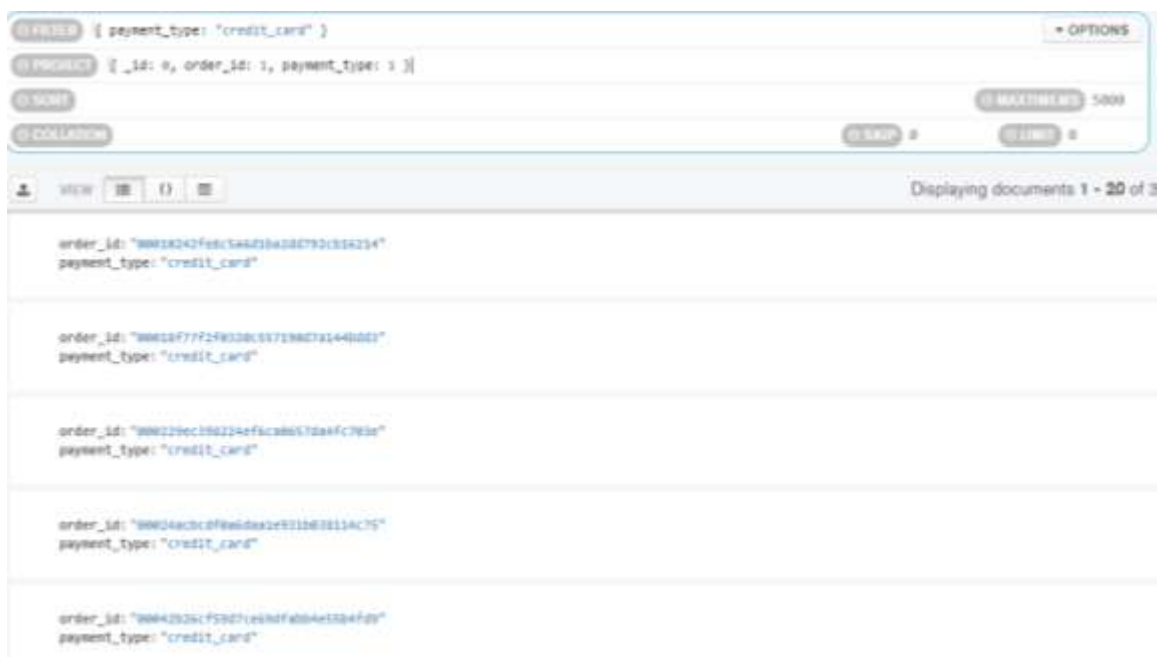
Sort: { shipping_limit_date: -1 }

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

	order_id	payment_type
▶	00010242fe8c5a6d1ba2dd792cb16214	credit_card
	00018f77f2f0320c557190d7a144bdd3	credit_card
	000229ec398224ef6ca0657da4fc703e	credit_card
	00024acbcdfoa6daa1e931b038114c75	credit_card
	00042b26cf59d7ce69dfabb4e55b4fd9	credit_card
	00054e8431b9d7675808bcb819fb4a32	credit_card

SENTENCIA: select order_id, payment_type from 4_olist_order_payments_dataset where payment_type like 'credit_card';



Agrupamientos y segmentaciones (Post Work II & V)

MySQL WORKBENCH

SENTENCIA: select count(*) from 2_olist_geolocation_dataset where geolocation_zip_code_prefix like '6%';

MONGODB

Proyecto.2_olist_geolocation_dataset

DOCUMENTS 500 TOTAL SIZE 98.2KB AVG SIZE 200B INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter: { geolocation_zip_code_prefix: /^6/ }

PROJECTION: SCHEMA: COLLATION: SHOWING 5000 SKIP: LIMIT:

ADD DATA VIEW

Displaying documents 1 - 20 of 497

```

{
  "_id": "ObjectID('5f9d2e29e89d740483c12ef')",
  "id": "778187",
  "geolocation_zip_code_prefix": "09948",
  "geolocation_lat": "-9.065326474",
  "geolocation_lng": "-68.66497279",
  "geolocation_city": "santa marta",
  "geolocation_state": "AC"
}

{
  "_id": "ObjectID('5f9d2e29e89d740483c12f0')",
  "id": "778188",
  "geolocation_zip_code_prefix": "09918",
  "geolocation_lat": "-9.935545",
  "geolocation_lng": "-47.83511725",
  "geolocation_city": "rio branco",
  "geolocation_state": "AC"
}

{
  "_id": "ObjectID('5f9d2e29e89d740483c12f1')",
  "id": "778189",
  "geolocation_zip_code_prefix": "09918",
  "geolocation_lat": "-9.935545",
  "geolocation_lng": "-47.83511725",
  "geolocation_city": "rio branco",
  "geolocation_state": "AC"
}

```

JSON: Filter: { geolocation_zip_code_prefix: /^6/ }

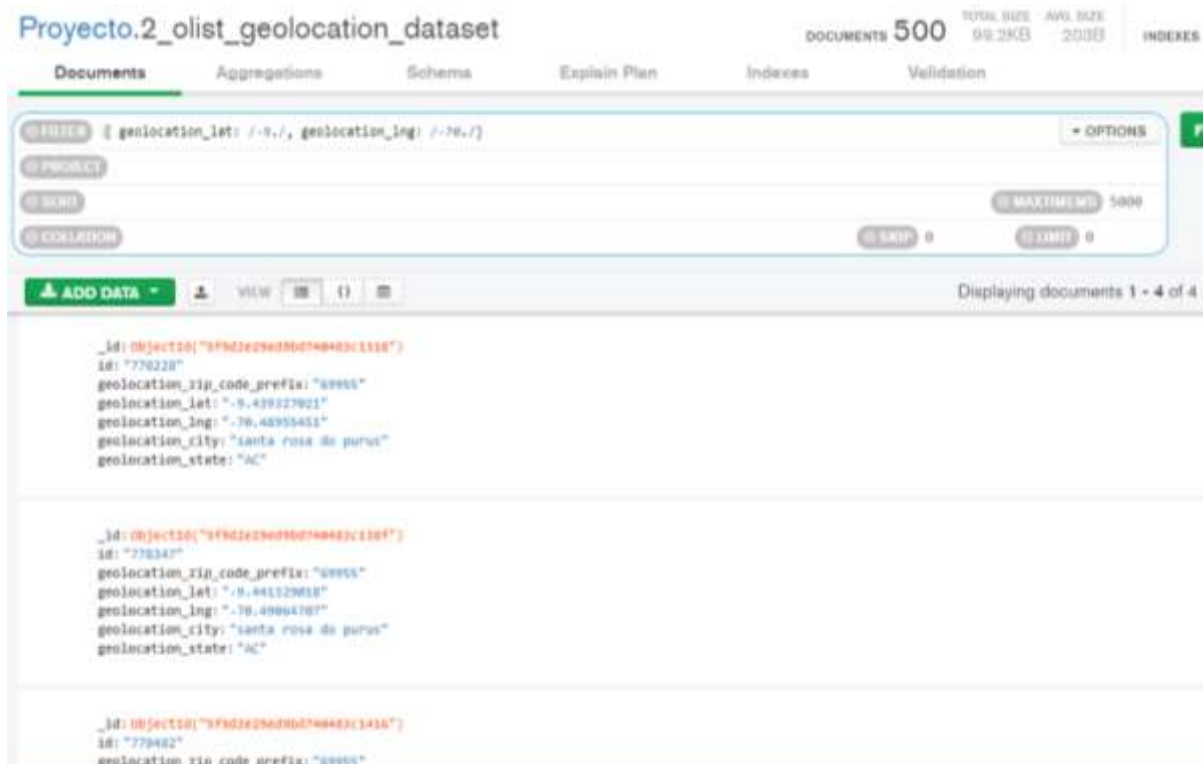
2) PATRONES. ¿Cuál es el total de aquellos id cuya latitud sea -9 y cuya longitud sea -70? (2)

MySQL WORKBENCH

	count(*)
▶	4

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

	id	avg(price)
▶	177	99
	360	99.5
	967	129
	76	78
	689	93
	331	125.9

Result 29 x

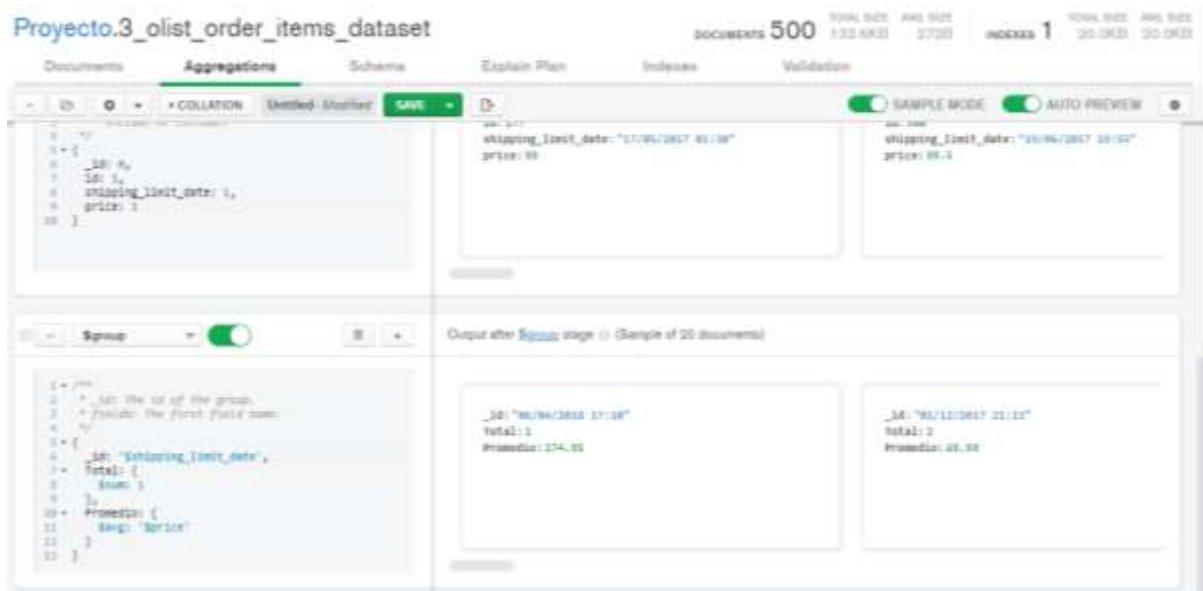
Output

Action Output

#	Time	Action	Message
✓ 65	00:20:52	select * from 3_olist_order_items_dataset	500 row(s) returned
✓ 66	00:21:04	select id, avg(price) from 3_olist_order_items_dataset group by shipping_limit_date	430 row(s) returned

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

	payment_type	count(order_id)
▶	credit_card	366
	boleto	97
	voucher	28
	debit_card	8
	not_defined	1

Result 34 ×

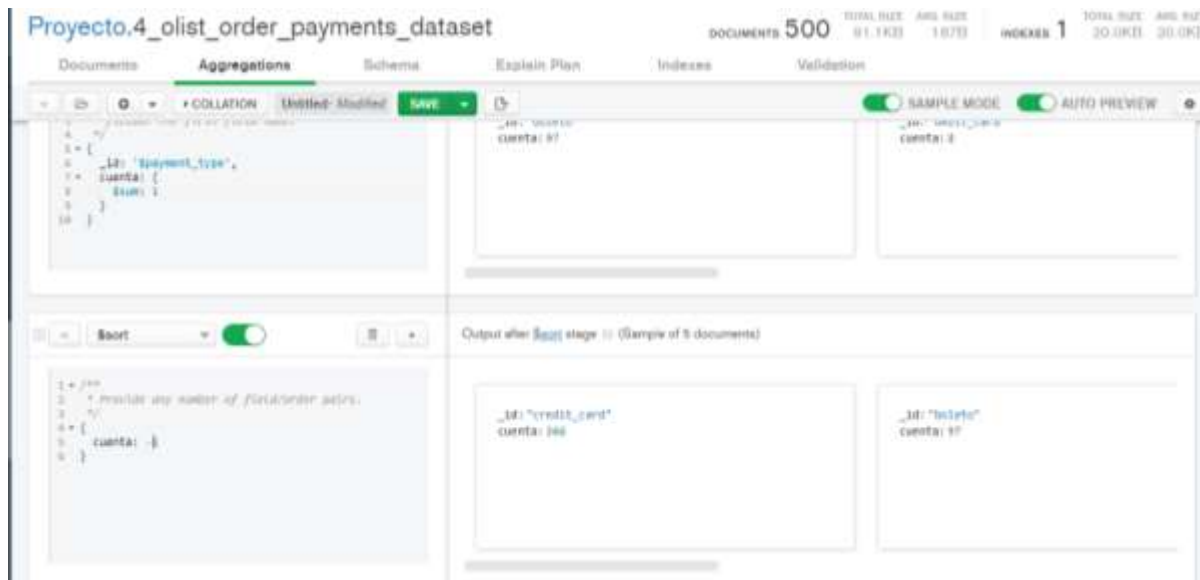
Output

Action Output

#	Time	Action	Message
70	00:24:03	select count(order_id) from 4_olist_order_payments_dataset group by payment_type	5 row(s) returned
71	00:24:19	select payment_type, count(order_id) from 4_olist_order_payments_dataset group by payment_type	5 row(s) returned

SENTENCIA: select payment_type, count(order_id) from
4olist_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

	Cantidad
▶	166

SENTENCIA: select count(payment_id) as Cantidad
from 4olist_order_payments_dataset
where payment_value > (select round(avg(payment_value), 2) as Promedio
from 4olist_order_payments_dataset);

MONGODB

Proyecto.4_olist_order_payments_dataset

DOCUMENTS 500 TOTAL SIZE 91.1 KB AVG SIZE 182 B INDEXES 1

Documents Aggregations Schema Explain Plan Indexes Validation

+ COLLATOR Unsaved Modified Save

SAMPLE MODE AUTO PREVIEW

```

1 {
2   //
3   // Aggregations: the fields to
4   // include or exclude.
5   //
6   //
7   //
8   //
9   //
10  //
11  //
12  //
13  //
14  //
15  //
16  //
17  //
18  //
19  //
20  //
21  //
22  //
23  //
24  //
25  //
26  //
27  //
28  //
29  //
30  //
31  //
32  //
33  //
34  //
35  //
36  //
37  //
38  //
39  //
40  //
41  //
42  //
43  //
44  //
45  //
46  //
47  //
48  //
49  //
50  //
51  //
52  //
53  //
54  //
55  //
56  //
57  //
58  //
59  //
60  //
61  //
62  //
63  //
64  //
65  //
66  //
67  //
68  //
69  //
70  //
71  //
72  //
73  //
74  //
75  //
76  //
77  //
78  //
79  //
80  //
81  //
82  //
83  //
84  //
85  //
86  //
87  //
88  //
89  //
90  //
91  //
92  //
93  //
94  //
95  //
96  //
97  //
98  //
99  //
100 //

```

Output after **Project** stage (Sample of 1 document)

```

1 {
2   //
3   // Aggregations: the fields to
4   // include or exclude.
5   //
6   //
7   //
8   //
9   //
10  //
11  //
12  //
13  //
14  //
15  //
16  //
17  //
18  //
19  //
20  //
21  //
22  //
23  //
24  //
25  //
26  //
27  //
28  //
29  //
30  //
31  //
32  //
33  //
34  //
35  //
36  //
37  //
38  //
39  //
40  //
41  //
42  //
43  //
44  //
45  //
46  //
47  //
48  //
49  //
50  //
51  //
52  //
53  //
54  //
55  //
56  //
57  //
58  //
59  //
60  //
61  //
62  //
63  //
64  //
65  //
66  //
67  //
68  //
69  //
70  //
71  //
72  //
73  //
74  //
75  //
76  //
77  //
78  //
79  //
80  //
81  //
82  //
83  //
84  //
85  //
86  //
87  //
88  //
89  //
90  //
91  //
92  //
93  //
94  //
95  //
96  //
97  //
98  //
99  //
100 //

```

8_olist_sellers_dataset

Promedio

Vista

Proyecto.4_olist_order_payments_dataset

DOCUMENTS 500 TOTAL SIZE 91.1 KB AVG SIZE 182 B INDEXES 1

Documents Aggregations Schema Explain Plan Indexes Validation

+ COLLATOR Unsaved Modified Save

SAMPLE MODE AUTO PREVIEW

```

1 {
2   //
3   // Aggregations: the fields to
4   // include or exclude.
5   //
6   //
7   //
8   //
9   //
10  //
11  //
12  //
13  //
14  //
15  //
16  //
17  //
18  //
19  //
20  //
21  //
22  //
23  //
24  //
25  //
26  //
27  //
28  //
29  //
30  //
31  //
32  //
33  //
34  //
35  //
36  //
37  //
38  //
39  //
40  //
41  //
42  //
43  //
44  //
45  //
46  //
47  //
48  //
49  //
50  //
51  //
52  //
53  //
54  //
55  //
56  //
57  //
58  //
59  //
60  //
61  //
62  //
63  //
64  //
65  //
66  //
67  //
68  //
69  //
70  //
71  //
72  //
73  //
74  //
75  //
76  //
77  //
78  //
79  //
80  //
81  //
82  //
83  //
84  //
85  //
86  //
87  //
88  //
89  //
90  //
91  //
92  //
93  //
94  //
95  //
96  //
97  //
98  //
99  //
100 //

```

Output after **Count** stage (Sample of 1 document)

```

1 {
2   //
3   // Aggregations: the fields to
4   // include or exclude.
5   //
6   //
7   //
8   //
9   //
10  //
11  //
12  //
13  //
14  //
15  //
16  //
17  //
18  //
19  //
20  //
21  //
22  //
23  //
24  //
25  //
26  //
27  //
28  //
29  //
30  //
31  //
32  //
33  //
34  //
35  //
36  //
37  //
38  //
39  //
40  //
41  //
42  //
43  //
44  //
45  //
46  //
47  //
48  //
49  //
50  //
51  //
52  //
53  //
54  //
55  //
56  //
57  //
58  //
59  //
60  //
61  //
62  //
63  //
64  //
65  //
66  //
67  //
68  //
69  //
70  //
71  //
72  //
73  //
74  //
75  //
76  //
77  //
78  //
79  //
80  //
81  //
82  //
83  //
84  //
85  //
86  //
87  //
88  //
89  //
90  //
91  //
92  //
93  //
94  //
95  //
96  //
97  //
98  //
99  //
100 //

```

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

	Field	Type	Null	Key	Default	Extra
►	seller_id	text	YES		NULL	
	seller_zip_code_prefix	int	YES		NULL	
	seller_city	text	YES		NULL	
	seller_state	text	YES		NULL	

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)

The screenshot shows the MongoDB Compass interface. At the top, it displays 'Proyecto.3_olist_order_items_dataset' with 500 documents, 1.32 MB size, and 1 index. The 'Aggregations' tab is active, showing a pipeline with a \$lookup stage. The \$lookup stage is configured with 'from: '8_olist_sellers_dataset'', 'localField: 'product_id'', 'foreignField: 'product_id'', and 'as: 'Union''. The output shows a sample of 20 documents, each containing the original order item fields plus a new 'Union' array field containing the seller information.

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

	Field	Type	Null	Key	Default	Extra
►	customer_id	text	YES		NULL	
	customer_unique_id	text	YES		NULL	
	customer_zip_code_prefix	int	YES		NULL	
	customer_city	text	YES		NULL	
	customer_state	text	YES		NULL	

	Field	Type	Null	Key	Default	Extra
▶	id	int	YES		NULL	
	geolocation_zip_code_prefix	int	YES		NULL	
	geolocation_lat	double	YES		NULL	
	geolocation_lng	double	YES		NULL	
	geolocation_city	text	YES		NULL	
	geolocation_state	text	YES		NULL	



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)

The screenshot shows the MongoDB Compass interface with the 'Aggregations' tab selected. The pipeline is as follows:

```

1. $coll: {
  from: 'customers',
  localField: 'customer_zip_code_prefix',
  foreignField: 'zip_code_prefix',
  as: 'locations'
}
2. $lookup: {
  from: 'locations',
  localField: 'customer_zip_code_prefix',
  foreignField: 'zip_code_prefix',
  as: 'locations'
}

```

The output after the \$lookup stage shows a sample of 20 documents. The first document is:

```

{
  "_id": "50505050505050505050505050505050",
  "customer_id": "50505050505050505050505050505050",
  "customer_unique_id": "50505050505050505050505050505050",
  "customer_zip_code_prefix": "50505",
  "customer_city": "50505",
  "customer_state": "50505",
  "locations": [
    {
      "_id": "50505050505050505050505050505050",
      "customer_id": "50505050505050505050505050505050",
      "customer_unique_id": "50505050505050505050505050505050",
      "customer_zip_code_prefix": "50505",
      "customer_city": "50505",
      "customer_state": "50505"
    }
  ]
}

```

```
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

	customer_id	order_id	review_id
▶	004a2b3e8b026284407c620f4f01f686	01e71f17355fc56620b0d3678f796c42	NULL
	004d41e9bf012c669db1a257888d85a2	NULL	NULL
	004df18653e9438571e9d294776a5c5c	NULL	NULL
	004ee20df425002ed78553e1f50caa3e	NULL	NULL
	004f39eb181d53fa796fe82ce7a0cbd5	NULL	NULL
	00502500a53243e07961383bf1a6fa82	NULL	NULL

Result 91 x

Output

Action Output

#	Time	Action	Message
✓ 241	02:45:37	describe 2_olist_geolocation_dataset	6 row(s) returned
✓ 242	02:46:19	select a.customer_id, b.order_id, c.review_id from 1_olist_customers_dataset a left join 6_ol...	1000 row(s) returned

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

The screenshot shows the MongoDB Compass application. On the left, there's a sidebar with a list of collections under the 'Proyecto' database. The main area displays the 'Proyecto.1_olist_customers_dataset' collection. The 'Documents' tab is active, showing a list of documents. A sample document is expanded, showing its JSON structure:

```
{ "_id": 0, "order_id": 0, "customer_id": 0 }
```

. The interface also includes a 'Filter your data' section and a 'MongoDB Data' status bar at the bottom.

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)

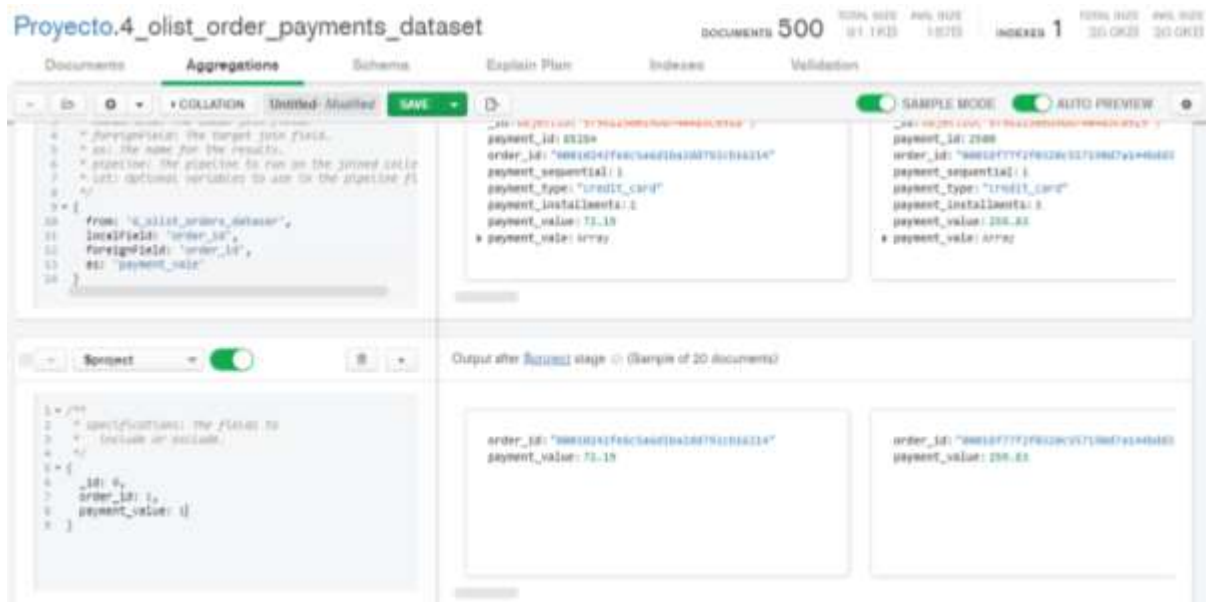
MySQL WORKBENCH

	order_id	payment_value
▶	010d14f76a229c295260e4ca4744acc7	44.69
	011375f941b3695c3c3ee58bfe0d3655	144.29
	00bd8cf08fa3998ea6ad6d16c2165822	34.28
	00891ba5de66f55000ee358ceea9b345	149.69
	00ba414462370af2d76195d12e0d5481	39.09
	009838529bb913846ab6670d22865381	176.4

#	Time	Action	Message
✓ 242	02:46:19	select a.customer_id, b.order_id, c.review_id from 1_olist_customers_dataset a left join 6_ol...	1000 row(s) returned
✓ 243	02:47:19	select a.order_id, b.payment_value from 6_olist_orders_dataset a left join 4_olist_order_pay...	258 row(s) returned

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

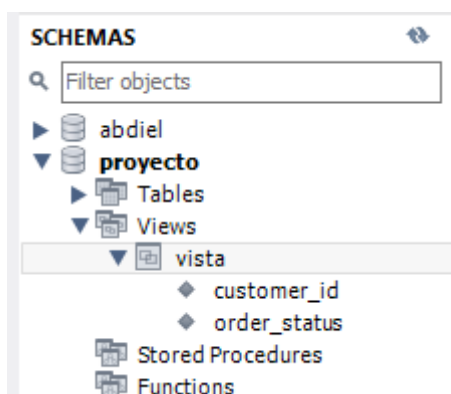


JSON:

```
{
  "$lookup": {
    "from": "6_olist_orders_dataset",
    "localField": "order_id",
    "foreignField": "order_id",
    "as": "payment_value"
  },
  "$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



	customer_id	order_status
▶	004a2b3e8b026284407c620f4f01f686	delivered
	010583f85dd55d32a6db1068f58ca6f1	delivered
	023d1d56f9187c91230a4e8b50ab6fb3	delivered
	0241bf6728f2c70f948e9a51484d1435	delivered
	02cb822eec8b6309e7ce715ccea3390d	delivered

Result 94 ×

Output :

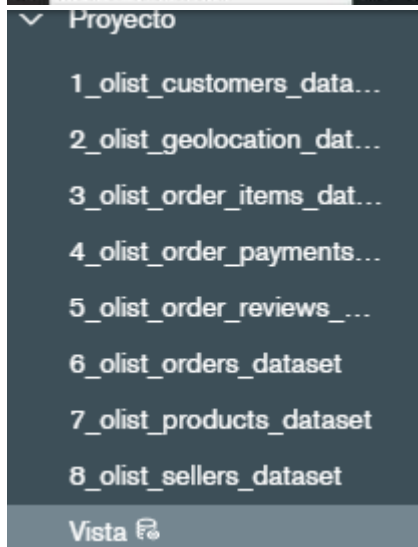
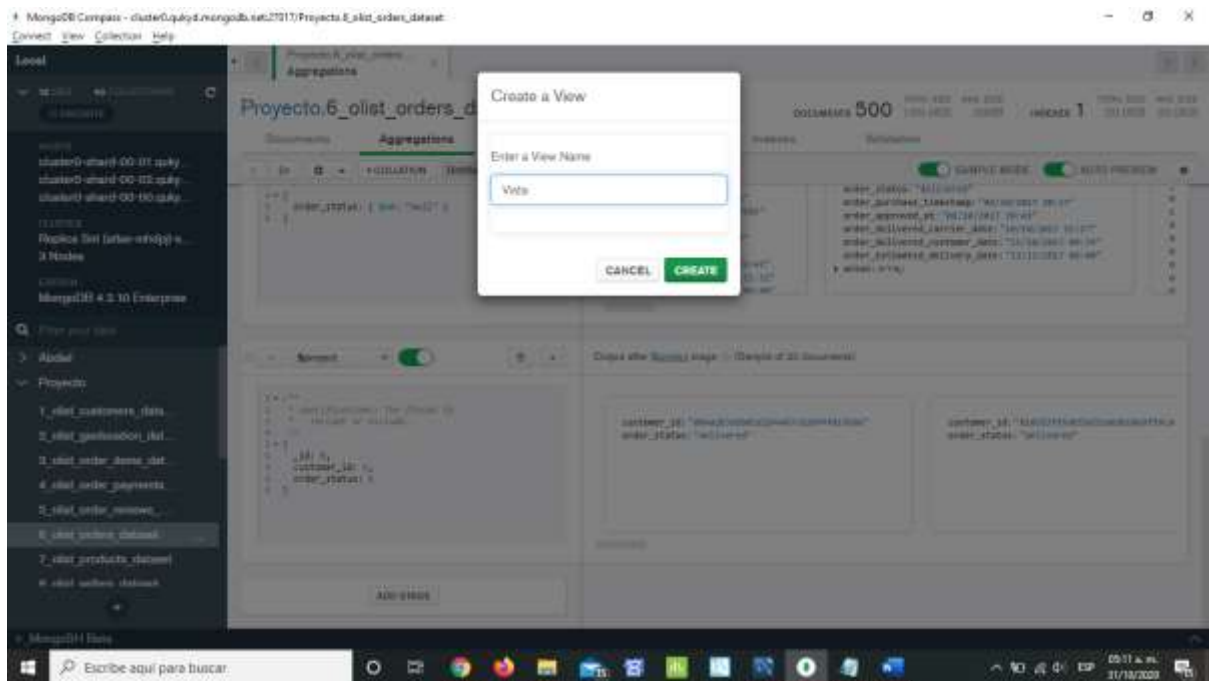
Action Output

#	Time	Action	Message
✓ 244	02:48:28	select a.customer_id, b.order_status from 1_olist_customers_dataset a left join 6_olist_order...	5 row(s) returned
✓ 245	02:48:31	select a.customer_id, b.order_status from 1_olist_customers_dataset a left join 6_olist_order...	5 row(s) returned

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

The screenshot shows the MongoDB Compass application. On the left, a sidebar lists collections under the 'Proyecto' database, including '1_olist_customers_data...', '2_olist_geolocation_dat...', '3_olist_order_items_dat...', '4_olist_order_payments...', '5_olist_order_reviews_', '6_olist_orders_dataset' (selected), '7_olist_products_dataset', and '8_olist_users_dataset'. The main panel displays the 'Proyecto.6_olist_orders_dataset' collection with 500 documents. It features a query editor with a JSON query: `{ "order_status": { "$ne": "Null" } }`. Below the query editor, there's a 'Preview' section showing a sample of 20 documents. The first document shown is: `{ "customer_id": "004a2b3e8b026284407c620f4f01f686", "order_status": "delivered" }`. The interface also includes a 'SAMPLE CODE' button and an 'AUTO PREVIEW' toggle.



JSON:

```
[{$lookup: {
  from: '1_olist_customers_data',
  localField: 'customer_id',
  foreignField: 'customer_id',
  as: 'union'
}}, {$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.