

AGOSTO 2020

---

**BEDU**

**N° Actividad:** Sesión 1 Fundamentos de SQL  
Sesión 2 Agrupaciones y subconsultas  
Sesión 3 Joins y Vistas  
Sesión 4 Fundamentos de MongoDB  
Sesión 5 Consultas en MongoDB  
Sesión 6 Agregaciones  
Sesión 7 Configuración de Bases de Datos Locales  
Sesión 8 Query competition

**Tema de la Actividad** Proyectos

Fecha de entrega 10-Ago-20

---

Data Analysis

Andrés Ramírez

---

José Adrián Lizardé Meléndez

Data-analysis-gdl-20-04

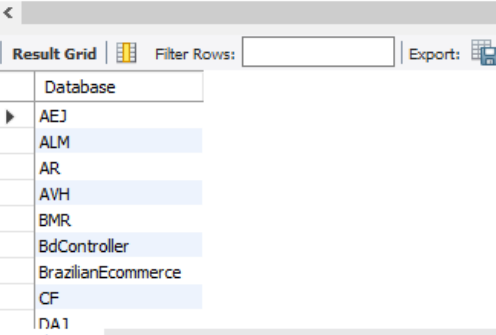
Ciudad de México.

---

## PROYECTO SESIÓN 1 FUNDAMENTOS DE SQL

1. Dentro del mismo servidor de bases de datos, conéctate al esquema classicmodels.

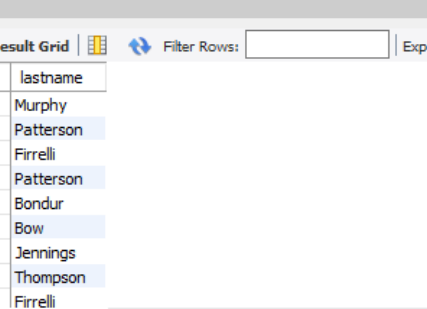
```
1 • show databases;
2 • use classicmodels;
3
```



Database
AEJ
ALM
AR
AVH
BMR
BdController
BrazilianEcommerce
CF
DA1

2. Dentro de la tabla employees, obtén el apellido de todos los empleados.

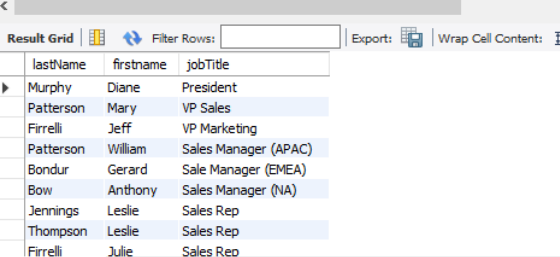
```
3 • select lastname from employees;
```



lastname
Murphy
Patterson
Firrelli
Patterson
Bondur
Bow
Jennings
Thompson
Firrelli

3. Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados.

```
4 • select lastName, firstName, jobTitle from employees;
```



lastName	firstName	jobTitle
Murphy	Diane	President
Patterson	Mary	VP Sales
Firrelli	Jeff	VP Marketing
Patterson	William	Sales Manager (APAC)
Bondur	Gerard	Sale Manager (EMEA)
Bow	Anthony	Sales Manager (NA)
Jennings	Leslie	Sales Rep
Thompson	Leslie	Sales Rep
Firrelli	Julie	Sales Rep

4. Dentro de la tabla employees, obtén todos los datos de cada empleado.

```
5 • select * from employees;
```

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	1002	President
1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)
1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep

5. Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan el puesto Sales Rep.

```
6 • select lastName, firstName, jobTitle
7 from employees
8 where jobTitle = "Sales Rep";
```

lastName	firstName	jobTitle
Jennings	Leslie	Sales Rep
Thompson	Leslie	Sales Rep
Firrelli	Julie	Sales Rep
Patterson	Steve	Sales Rep
Tseng	Foon Yue	Sales Rep
Vanauf	George	Sales Rep
Bondur	Loui	Sales Rep
Hernandez	Gerard	Sales Rep
Castillo	Pamela	Sales Rep

6. Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep y código de oficina 1.

```
9 • select lastName, firstName, officeCode
10 from employees
11 where jobTitle = "Sales Rep"
12 and officeCode = 1;
```

lastName	firstName	officeCode
Jennings	Leslie	1
Thompson	Leslie	1

7. Dentro de la tabla employees, obtén el apellido, nombre, puesto y código de oficina de todos los empleados que tengan el puesto Sales Rep o código de oficina 1.

```
9 • select lastName, firstName, officeCode
10 from employees
11 where jobTitle = "Sales Rep"
12 and officeCode = 1;
```

lastName	firstName	officeCode
Jennings	Leslie	1
Thompson	Leslie	1

8. Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados que tenga código de oficina 1, 2 o 3.

```
13 • select lastName, firstname, officeCode
14 from employees
15 where jobTitle = "Sales Rep"
16 AND officeCode IN (1,2,3);
```

lastName	firstname	officeCode
Jennings	Leslie	1
Thompson	Leslie	1
Firrelli	Julie	2
Patterson	Steve	2
Tseng	Foon Yue	3
Vanauf	George	3

9. Dentro de la tabla employees, obtén el apellido, nombre y puesto de todos los empleados que tengan un puesto distinto a Sales Rep.

```
13 • select lastName, firstname, jobTitle
14 from employees
15 where jobTitle <> "Sales Rep";
16
```

lastName	firstname	jobTitle
Murphy	Diane	President
Patterson	Mary	VP Sales
Firrelli	Jeff	VP Marketing
Patterson	William	Sales Manager (APAC)
Bondur	Gerard	Sale Manager (EMEA)
Bow	Anthony	Sales Manager (NA)

10. Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea mayor a 5.

```
16 • select lastName, firstname, officeCode
17 from employees
18 where jobTitle = "Sales Rep"
19 and officeCode > 5;
```

lastName	firstname	officeCode
Bott	Larry	7
Jones	Barry	7
Fixter	Andy	6
Marsh	Peter	6
King	Tom	6

11. Dentro de la tabla employees, obtén el apellido, nombre y código de oficina de todos los empleados cuyo código de oficina sea menor o igual 4.

```
39  /*11.  Dentro de la tabla employees, obtén el apellido, nombre y código
40  de oficina de todos los empleados cuyo código de oficina sea menor o igual 4.*/
41  •  select lastName, firstname, officeCode
42  from employees
43  where jobTitle = "Sales Rep"
44  and officeCode <= 4;
45
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
lastName	firstname	officeCode	
Jennings	Leslie	1	
Thompson	Leslie	1	
Firrelli	Julie	2	
Patterson	Steve	2	
Tseng	Foon Yue	3	
Vanauf	George	3	
Bondur	Loui	4	
Hernandez	Gerard	4	
Castillo	Pamela	4	

12. Dentro de la tabla customers, obtén el nombre, país y estado de todos los clientes cuyo país sea USA y cuyo estado sea CA.

```
46  /*12.  Dentro de la tabla customers, obtén el nombre, país y estado de todos los
47  clientes cuyo país sea USA y cuyo estado sea CA.*/
48  •  select customerName, country, state
49  from customers
50  where country = "USA"
51  and state = "CA";
52
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
customerName	country	state	
Mini Gifts Distributors Ltd.	USA	CA	
Mini Wheels Co.	USA	CA	
Technics Stores Inc.	USA	CA	
Toys4GrownUps.com	USA	CA	
Boards & Toys Co.	USA	CA	
Collectable Mini Designs Co.	USA	CA	
Corporate Gift Ideas Co.	USA	CA	
Men 'R' US Retailers, Ltd.	USA	CA	
The Sharp Gifts Warehouse	USA	CA	

13. Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos los clientes cuyo país sea, USA, cuyo estado sea CA y cuyo límite de crédito sea mayor a 100000.

```
53  /*13.  Dentro de la tabla customers, obtén el nombre, país, estado y límite de crédito de todos
54  los clientes cuyo país sea, USA, cuyo estado sea CA y cuyo límite de crédito sea mayor a 100000.*/
55  •  select customerName, country, state, creditLimit
56  from customers
57  where state = "CA"
58  and creditLimit > 100000;
59
```

<

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	customerName	country	state	creditLimit
▶	Mini Gifts Distributors Ltd.	USA	CA	210500.00
	Collectable Mini Designs Co.	USA	CA	105000.00
	Corporate Gift Ideas Co.	USA	CA	105000.00

14. Dentro de la tabla customers, obtén el nombre y país de todos los clientes cuyo país sea USA o France.

```
60  /*14.  Dentro de la tabla customers, obtén el nombre y país de todos los clientes cuyo
61  país sea USA o France.*/
62  •  select customerName, country
63  from customers
64  where country = "USA" or country = "France";
```

<

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	customerName	country
▶	Atelier graphique	France
	Signal Gift Stores	USA
	La Rochelle Gifts	France
	Mini Gifts Distributors Ltd.	USA
	Mini Wheels Co.	USA
	Land of Toys Inc.	USA
	Saveley & Henriot, Co.	France
	Muscle Machine Inc	USA
	Diecast Classics Inc.	USA

15. Dentro de la tabla customers, obtén el nombre, país y límite de crédito de todos los clientes cuyo país sea USA o France y cuyo límite de crédito sea mayor a 100000. Para este ejercicio ten cuidado con los paréntesis.

```
66  /*15.  Dentro de la tabla customers, obtén el nombre, país y límite de crédito de todos los
67  clientes cuyo país sea USA o France y cuyo límite de crédito sea mayor a 100000. Para este
68  ejercicio ten cuidado con los paréntesis.*/
69  •  select customerName, country, creditLimit
70  from customers
71  where country = "USA" or country = "France"
72  and creditlimit > 100000;
73
```

<

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	customerName	country	creditLimit
	Online Diecast Creations ...	USA	114200.00
	Collectable Mini Designs ...	USA	105000.00
	Corporate Gift Ideas Co.	USA	105000.00
	Mini Classics	USA	102700.00
	Diecast Classics Inc.	USA	100600.00

16. Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que se encuentren en USA o France.

```
74 /*16. Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas
75 oficinas que se encuentren en USA o France.*/
76 • Select officeCode, city, phone, country
77 from offices
78 where country = "USA" or country = "France";
```

Result Grid

	officeCode	city	phone	country
▶	1	San Francisco	+1 650 219 4782	USA
	2	Boston	+1 215 837 0825	USA
	3	NYC	+1 212 555 3000	USA
	4	Paris	+33 14 723 4404	France
*	NULL	NULL	NULL	NULL

17. Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas que *no* se encuentren en USA o France.

```
80 /*17. Dentro de la tabla offices, obtén el código de la oficina, ciudad, teléfono y país de aquellas oficinas
81 que no se encuentren en USA o France.*/
82 • Select officeCode, city, phone, country
83 from offices
84 where country <> "USA" and country <> "France";
```

Result Grid

	officeCode	city	phone	country
▶	5	Tokyo	+81 33 224 5000	Japan
	6	Sydney	+61 2 9264 2451	Australia
	7	London	+44 20 7877 2041	UK
*	NULL	NULL	NULL	NULL

18. Dentro de la tabla orders, obtén el número de orden, número de cliente, estado y fecha de envío de todas las órdenes con el número 10165, 10287 o 10310.

```
86 /*18. Dentro de la tabla orders, obtén el número de orden, número de cliente, estado y fecha de envío de todas las
87 órdenes con el número 10165, 10287 o 10310.*/
88 • select orderNumber, customerNumber, status, shippedDate
89 from orders
90 where orderNumber IN (10165, 10287, 10310);
91
```

Result Grid

	orderNumber	customerNumber	status	shippedDate
▶	10165	148	Shipped	2003-12-26
	10287	298	Shipped	2004-09-01
	10310	259	Shipped	2004-10-18
*	NULL	NULL	NULL	NULL

19. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma ascendente.

```
92 /*19. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados
93 por apellido de forma ascendente.*/
94 • select contactLastName, contactFirstName
95 from customers
96 order by contactLastName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	contactLastName	contactFirstName
▶	Accorti	Paolo
	Altagar,G M	Raanan
	Andersen	Mel
	Anton	Carmen
	Ashworth	Rachel
	Barajas	Miguel
	Benitez	Violeta
	Bennett	Helen
	Berdalund	Christina

Result Grid  
Form Editor

20. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma descendente.

```
98 /*20. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados
99 por apellido de forma descendente.*/
100 • select contactLastName, contactFirstName
101 from customers
102 order by contactLastName desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	contactLastName	contactFirstName
▶	Young	Jeff
	Young	Julie
	Young	Mary
	Young	Dorothy
	Yoshido	Juri
	Walker	Brydey
	Victorino	Wendy
	Urs	Braun
	Tsena	Jerrv

21. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados por apellido de forma descendente y luego por nombre de forma ascendente.

```
104 /*21. Dentro de la tabla customers, obtén el apellido y nombre de cada cliente y ordena los resultados
105 por apellido de forma descendente y luego por nombre de forma ascendente.*/
106 • select contactLastName, contactFirstName
107 from customers
108 order by contactLastName desc, contactFirstName asc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	contactLastName	contactFirstName
▶	Young	Dorothy
	Young	Jeff
	Young	Julie
	Young	Mary
	Yoshido	Juri
	Walker	Brydey
	Victorino	Wendy
	Urs	Braun
	Tsena	Jerrv



22. Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más alto (top 5).

```
110 /*22. Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de
111 crédito de los cinco clientes con el límite de crédito más alto (top 5).*/
112 • select customerNumber, customerName, creditLimit
113 from customers
114 order by creditLimit desc
115 limit 5;
```

	customerNumber	customerName	creditLimit
▶	141	Euro+ Shopping Channel	227600.00
	124	Mini Gifts Distributors Ltd.	210500.00
	298	Vida Sport, Ltd	141300.00
	151	Muscle Machine Inc	138500.00
	187	AV Stores, Co.	136800.00
*	NULL	NULL	NULL

23. Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de crédito de los cinco clientes con el límite de crédito más bajo.

```
117 /*23. Dentro de la tabla customers, obtén el número de cliente, nombre de cliente y el límite de
118 crédito de los cinco clientes con el límite de crédito más bajo.*/
119 • select customerNumber, customerName, creditLimit
120 from customers
121 order by creditLimit
122 limit 5;
```

	customerNumber	customerName	creditLimit
▶	223	Natürlich Autos	0.00
	168	American Souvenirs Inc	0.00
	169	Porto Imports Co.	0.00
	206	Asian Shopping Network, Co	0.00
	125	Havel & Zbyszek Co	0.00
*	NULL	NULL	NULL

## SESIÓN 2 AGRUPACIONES Y SUBCONSULTAS

1. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre empiece con a.

```
1 • use classicmodels;
2 /*1. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de
3 todos los empleados cuyo nombre empiece con a.*/
4 • select employeeNumber, lastName, firstName
5 from employees
6 where firstName Like "a%";
```

	employeeNumber	lastName	firstName
▶	1143	Bow	Anthony
	1611	Fixter	Andy
*	NULL	NULL	NULL

2. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre termina con on.

```
8 /*2. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de
9 todos los empleados cuyo nombre termina con on.*/
10 • select employeeNumber, lastName, firstName
11 from employees
12 where firstName like "%on";
13
```

	employeeNumber	lastName	firstName
*	NULL	NULL	NULL

3. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de todos los empleados cuyo nombre incluye la cadena on.

```
14 /*3. Dentro de la tabla employees, obten el número de empleado, apellido y nombre de
15 todos los empleados cuyo nombre incluye la cadena on.*/
16 • select employeeNumber, lastName, firstName
17 from employees
18 where firstName like "%on%";
19
```

	employeeNumber	lastName	firstName
▶	1143	Bow	Anthony
	1286	Tseng	Foon Yue
*	NULL	NULL	NULL

4. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyos nombres tienen tres letras e inician con T y finalizan con m.\*

```
20 /*4. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de
21 todos los empleados cuyos nombres tienen tres letras e inician con T y finalizan con m.*
22 • select employeeNumber, lastName, firstName
23 from employees
24 where length(firstName)=3
25 and firstName LIKE "t%"
26 and firstName LIKE "%m";
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
employeeNumber	lastName	firstName		
1619	King	Tom		
NULL	NULL	NULL		

5. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de todos los empleados cuyo nombre *no* inicia con B.

```
28 /*5. Dentro de la tabla employees, obtén el número de empleado, apellido y nombre de
29 todos los empleados cuyo nombre no inicia con B.*/
30 • select employeeNumber, lastName, firstName
31 from employees
32 where firstName not like "b%";
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
employeeNumber	lastName	firstName		
1002	Murphy	Diane		
1056	Patterson	Mary		
1076	Firrelli	Jeff		
1088	Patterson	William		
1102	Bondur	Gerard		
1143	Bow	Anthony		
1165	Jennings	Leslie		
1166	Thompson	Leslie		
1188	Firrelli	Julie		

6. Dentro de la tabla products, obtén el código de producto y nombre de los productos cuyo código incluye la cadena \_20.

```
34 /*6. Dentro de la tabla products, obtén el código de producto y nombre de los productos
35 cuyo código incluye la cadena _20.*/
36 • select productCode, productName
37 from products
38 where productCode like '%_20';
39
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
productCode	productName			
S18_3320	1917 Maxwell Touring Car			
S24_3420	1937 Horch 930V Limousine			
S24_4620	1961 Chevrolet Impala			
NULL	NULL			

7. Dentro de la tabla orderdetails, obten el total de cada orden.

```
39
40  /*7.    Dentro de la tabla orderdetails, obten el total de cada orden.*/
41  •  select orderNumber, ROUND(SUM(total), 2) as total_orden
42     from (select orderNumber, quantityOrdered * priceEach as total
43           from orderdetails) as subquery
44     group by orderNumber
45     order by orderNumber;
46
```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	orderNumber	total_orden
▶	10100	10223.83
	10101	10549.01
	10102	5494.78
	10103	50218.95
	10104	40206.20
	10105	53959.21
	10106	52151.81
	10107	22292.62
	10108	51001.22

8. Dentro de la tabla orders obten el número de órdenes por año.

```
47  /*8.    Dentro de la tabla orders obten el número de órdenes por año*/
48  •  select EXTRACT(year from orderDate) as año, COUNT(*)
49     from orders
50     group by año;
51
```



<

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	año	COUNT(*)
▶	2003	111
	2004	151
	2005	64







9. Obten el apellido y nombre de los empleados cuya oficina está ubicada en USA.

```
52  /*9.  Obten el apellido y nombre de los empleados cuya oficina está ubicada en USA.*/
53  •  select lastName, firstName
54      from employees
55  where officeCode in (
56      select officeCode
57      from offices
58      where country like 'USA');
59
```

Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	lastName	firstName
▶	Murphy	Diane
	Patterson	Mary
	Firrelli	Jeff
	Bow	Anthony
	Jennings	Leslie
	Thompson	Leslie
	Firrelli	Julie
	Patterson	Steve
	Tseng	Foon Yue

10. Obten el número de cliente, número de cheque y cantidad del cliente que ha realizado el pago más alto.

```
60  /*10.  Obten el número de cliente, número de cheque y cantidad del cliente que ha
61  realizado el pago más alto.*/
62  •  select customerNumber, checkNumber, amount
63      from payments
64  where amount = (select MAX(amount)
65      from payments);
66
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:    Export/Import:   Wrap Cell Content: 			
	customerNumber	checkNumber	amount
▶	141	JE105477	120166.58
*	NULL	NULL	NULL

11. Obten el número de cliente, número de cheque y cantidad de aquellos clientes cuyo pago es más alto que el promedio.

```
67  /*11.  Obten el número de cliente, número de cheque y cantidad de aquellos clientes
68  cuyo pago es más alto que el promedio.*/
69  •  select customerNumber, checkNumber, amount
70  from payments
71  where amount > (select avg(amount)
72  from payments);
73
```

	customerNumber	checkNumber	amount
▶	112	HQ55022	32641.98
	112	ND748579	33347.88
	114	GG31455	45864.03
	114	MA765515	82261.22
	114	NR27552	44894.74
	119	LN373447	47924.19
	119	NG94694	49523.67
	121	DB889831	50218.95
	121	MA307151	34638.14

12. Obten el nombre de aquellos clientes que no han hecho ninguna orden.

```
74  /*12.  Obten el nombre de aquellos clientes que no han hecho ninguna orden.*/
75  •  select customerNumber
76  from customers
77  where customerNumber not in (select distinct(customerNumber)
78  from payments);
79
```

	customerNumber
▶	125
	169
	206
	223
	237
	247
	273
	293
	303

13. Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.

```
80  /*13.  Obten el máximo, mínimo y promedio del número de productos en las órdenes de venta.*/
81  •  SELECT AVG(cuenta) AS prom_productos, MAX(cuenta) AS max_productos, MIN(cuenta) AS min_productos
82  FROM (SELECT orderNumber, COUNT(*) as cuenta
83  FROM orderdetails
84  GROUP BY orderNumber) AS sq;
85
```

	prom_productos	max_productos	min_productos
▶	9.1902	18	1

14. Dentro de la tabla orders, obten el número de órdenes que hay por cada estado.

```
86  /*14.  Dentro de la tabla orders, obten el número de órdenes que hay por cada estado.*/
87  •  select estado, SUM(cuenta) as ordenes_estado
88  from (select customerNumber, COUNT(*) as cuenta, (select c.state from customers c
89  where c.customerNumber = o.customerNumber) as estado
90  from orders o
91  group by estado, 1) sq
92  group by estado
93  order by 2;
```

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

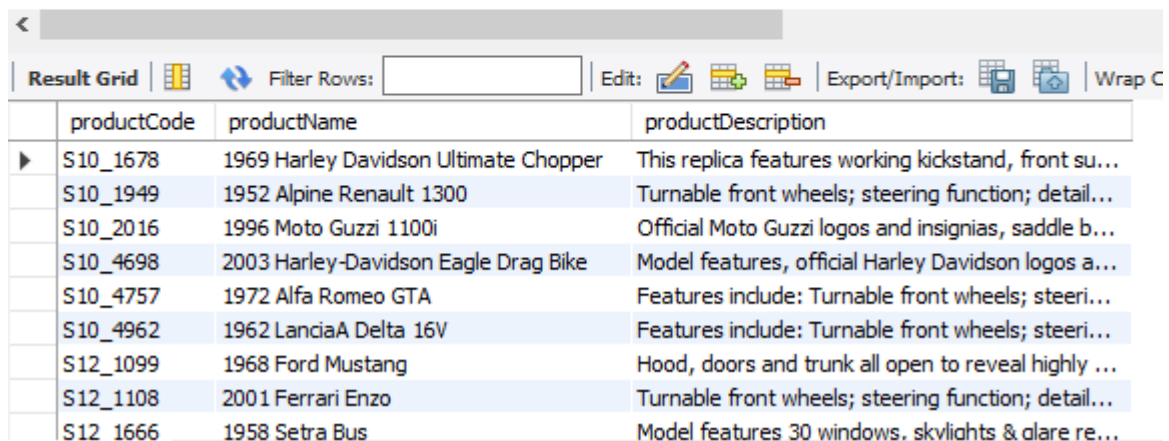
	estado	ordenes_estado
▶	Isle of Wight	2
	Osaka	2
	NV	3
	NH	3
	Queensland	3
	NJ	3
	Québec	3
	BC	4

## SESIÓN 3 JOINS Y VISTAS

Para estas consultas usa *RIGHT JOIN*

1. Obten el código de producto, nombre de producto y descripción de todos los productos.

```
1 SELECT productCode, productName, productDescription
2 FROM products;
3
```

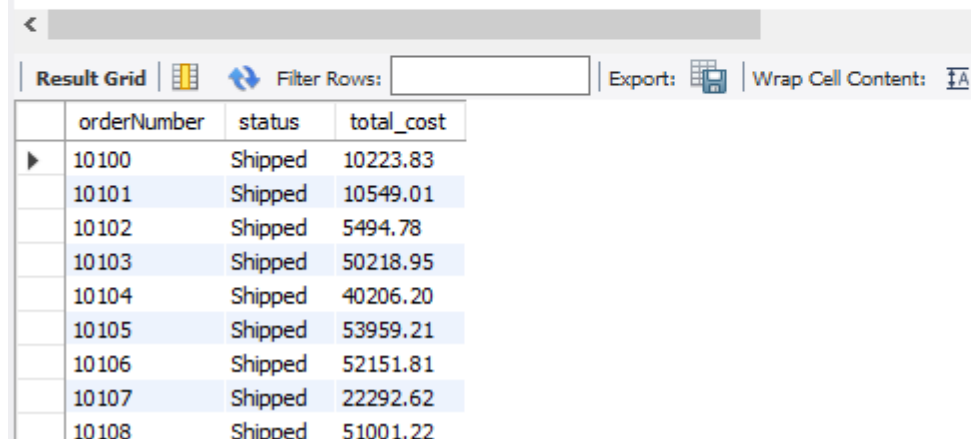


The screenshot shows a database query result grid with the following columns: productCode, productName, and productDescription. The results are as follows:

productCode	productName	productDescription
S10_1678	1969 Harley Davidson Ultimate Chopper	This replica features working kickstand, front su...
S10_1949	1952 Alpine Renault 1300	Turnable front wheels; steering function; detail...
S10_2016	1996 Moto Guzzi 1100i	Official Moto Guzzi logos and insignias, saddle b...
S10_4698	2003 Harley-Davidson Eagle Drag Bike	Model features, official Harley Davidson logos a...
S10_4757	1972 Alfa Romeo GTA	Features include: Turnable front wheels; steeri...
S10_4962	1962 LanciaA Delta 16V	Features include: Turnable front wheels; steeri...
S12_1099	1968 Ford Mustang	Hood, doors and trunk all open to reveal highly ...
S12_1108	2001 Ferrari Enzo	Turnable front wheels; steering function; detail...
S12_1666	1958 Setra Bus	Model features 30 windows, skvlights & clare re...

2. Obten el número de orden, estado y costo total de cada orden.

```
5 /*2. Obten el número de orden, estado y costo total de cada orden.*/
6 • SELECT od.orderNumber, o.status, SUM(od.quantityOrdered * od.priceEach) AS total_cost
7 FROM orders o RIGHT JOIN orderdetails od
8 ON o.orderNumber = od.orderNumber
9 GROUP BY od.orderNumber
10 ORDER BY od.orderNumber;
11
```



The screenshot shows a database query result grid with the following columns: orderNumber, status, and total\_cost. The results are as follows:

orderNumber	status	total_cost
10100	Shipped	10223.83
10101	Shipped	10549.01
10102	Shipped	5494.78
10103	Shipped	50218.95
10104	Shipped	40206.20
10105	Shipped	53959.21
10106	Shipped	52151.81
10107	Shipped	22292.62
10108	Shipped	51001.22



3. Obten el número de orden, fecha de orden, línea de orden, nombre del producto, cantidad ordenada y precio de cada pieza que muestre los detalles de cada orden.

```
12  /*3.  Obten el número de orden, fecha de orden, línea de orden, nombre del producto,
13  cantidad ordenada y precio de cada pieza que muestre los detalles de cada orden.*/
14  •  SELECT od.orderNumber, o.orderDate, od.orderLineNumber,
15  p.productName, od.quantityOrdered, od.priceEach
16  FROM orderdetails od JOIN products p ON od.productCode = p.productCode
17  RIGHT JOIN orders o ON od.orderNumber = o.orderNumber
18  ORDER BY 1, 3;
19
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	orderNumber	status	total_cost
▶	10100	Shipped	10223.83
	10101	Shipped	10549.01
	10102	Shipped	5494.78
	10103	Shipped	50218.95
	10104	Shipped	40206.20
	10105	Shipped	53959.21
	10106	Shipped	52151.81
	10107	Shipped	22292.62
	10108	Shipped	51001.22

4. Obtén el número de orden, nombre del producto, el precio sugerido de fábrica (msrp) y precio de cada pieza.

```
20  /*4.  Obtén el número de orden, nombre del producto, el precio sugerido de fábrica
21  (msrp) y precio de cada pieza.*/
22  •  SELECT od.orderNumber AS order_number, p.productName AS prod_name, p.MSRP
23  AS price_suggested, od.priceEach AS unit_price
24  FROM products p
25  RIGHT JOIN orderdetails od
26  ON p.productCode = od.productCode
27  ORDER BY 1,2;
28
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	orderNumber	status	total_cost
▶	10100	Shipped	10223.83
	10101	Shipped	10549.01
	10102	Shipped	5494.78
	10103	Shipped	50218.95
	10104	Shipped	40206.20
	10105	Shipped	53959.21
	10106	Shipped	52151.81
	10107	Shipped	22292.62

Para estas consultas usa *LEFT JOIN*

5. Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.

```
29  /*5.  Obtén el número de cliente, nombre de cliente, número de orden y estado de cada cliente.*/
30  •  SELECT o.customerNumber, c.customerName, o.orderNumber, c.state
31  FROM orders o LEFT JOIN customers c
32  ON o.customerNumber = c.customerNumber
33  ORDER BY o.customerNumber, o.orderNumber;
```

	customerNumber	customerName	orderNumber	state
▶	103	Atelier graphique	10123	NULL
	103	Atelier graphique	10298	NULL
	103	Atelier graphique	10345	NULL
	112	Signal Gift Stores	10124	NV
	112	Signal Gift Stores	10278	NV
	112	Signal Gift Stores	10346	NV
	114	Australian Collectors, Co.	10120	Victoria
	114	Australian Collectors, Co.	10125	Victoria

6. Obtén los clientes que no tienen una orden asociada.

```
35  /*6.  Obtén los clientes que no tienen una orden asociada.*/
36  •  SELECT c.customerNumber, c.customerName
37  FROM customers c LEFT JOIN orders o
38  ON o.customerNumber = c.customerNumber
39  WHERE o.orderNumber IS NULL
40  ORDER BY o.customerNumber;
```

	customerNumber	customerName
▶	348	Asian Treasures, Inc.
	481	Raanan Stores, Inc
	480	Kremlin Collectables, Co.
	477	Mit Vergnügen & Co.
	465	Anton Designs, Ltd.
	459	Warburg Exchange
	443	Feuer Online Stores, Inc
	409	Stuttgart Collectable Exchange

7. Obtén el apellido de empleado, nombre de empleado, nombre de cliente, número de cheque y total, es decir, los clientes asociados a cada empleado.

```
42  /*7.  Obtén el apellido de empleado, nombre de empleado, nombre de cliente,  
43  número de cheque y total, es decir, los clientes asociados a cada empleado.*/  
44  ●  SELECT e.lastName, e.firstName, c.customerName, p.checkNumber, p.amount  
45  FROM payments p RIGHT JOIN customers c ON p.customerNumber = c.customerNumber  
46  RIGHT JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber  
47  ORDER BY e.lastName, e.firstName;  
48
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	customerNumber	customerName		
▶	348	Asian Treasures, Inc.		
	481	Raanan Stores, Inc		
	480	Kremlin Collectables, Co.		
	477	Mit Vergnügen & Co.		
	465	Anton Designs, Ltd.		
	459	Warburg Exchange		
	443	Feuer Online Stores, Inc		
	409	Stuttgart Collectable Exchange		

## PROYECTO SESIÓN 4 FUNDAMENTOS DE MONGODB

1. Obtén los datos de contacto de cada compañía.

```
{
  project: {
    name: 1,
    email_address: 1,
    phone_number: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. The query editor at the top has the following configuration:

- FILTER:** (empty)
- PROJECT:** `{name: 1, email_address: 1, phone_number: 1}`
- SORT:** (empty)
- COLLATION:** (empty)
- MAXTIMEMS:** 5000
- SKIP:** 0
- LIMIT:** 0

Buttons: **FIND**, **RESET**, **...**

Below the query editor, the results are displayed in a table view. The status bar indicates "Displaying documents 1 - 20 of 9500".

Document
<pre>{   "_id": ObjectId("52cdef7c4bab8bd675297d8a"),   "name": "Wetpaint",   "email_address": "info@wetpaint.com",   "phone_number": "206.859.6300" }</pre>
<pre>{   "_id": ObjectId("52cdef7c4bab8bd675297d8e"),   "name": "Facebook",   "email_address": "",   "phone_number": "" }</pre>
<pre>{   "_id": ObjectId("52cdef7c4bab8bd675297d8f"),   "name": "OmniDrive",   "email_address": "info@omnidrive.com",   "phone_number": "660-675-5052" }</pre>

2. Obtén la fuente de cada tweet.

```
{
  project: {
    _id: 1,
    source: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. The query editor at the top has the following configuration:

- FILTER:** (empty)
- PROJECT:** `{_id: 1, source: 1}`
- SORT:** (empty)
- COLLATION:** (empty)
- MAXTIMEMS:** 5000
- SKIP:** 0
- LIMIT:** 0

Buttons: **FIND**, **RESET**, **...**

Below the query editor, the results are displayed in a table view. The status bar indicates "Displaying documents 1 - 20 of 24832".

Document
<pre>{   "_id": ObjectId("5c8eccb0caa187d17ca623f5"),   "source": "web" }</pre>
<pre>{   "_id": ObjectId("5c8eccb0caa187d17ca623f7"),   "source": "&lt;a href='http://www.tweetdeck.com' rel='nofollow'&gt;TweetDeck&lt;/a&gt;" }</pre>
<pre>{   "_id": ObjectId("5c8eccb0caa187d17ca623fa"),   "source": "&lt;a href='http://blackberry.com/twitter' rel='nofollow'&gt;Twitter for Bla..." }</pre>

3. Obtén el nombre de todas las compañías fundadas en octubre.

```
{
  filter: {
    founded_month: 10
  },
  project: {
    name: 1,
    founded_month: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. The query editor at the top contains the following JSON query:

```
{
  filter: {
    founded_month: 10
  },
  project: {
    name: 1,
    founded_month: 1
  }
}
```

Below the query editor, the results are displayed in a table. The table shows three documents, each with the following fields:

- `_id`: ObjectId("52cdef7c4bab8bd675297d8a")
- `name`: "Wetpaint"
- `founded_month`: 10

The interface also includes a "VIEW" button, a "DISPLAYING DOCUMENTS 1 - 20 OF 301" status bar, and a "REFRESH" button.

4. Obtén el nombre de todas las compañías fundadas en 2008.

```
{
  filter: {
    founded_year: 2008
  },
  project: {
    name: 1,
    founded_year: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. The query editor at the top contains the following JSON query:

```
{
  filter: {
    founded_year: 2008
  },
  project: {
    name: 1,
    founded_year: 1
  }
}
```

Below the query editor, the results are displayed in a table. The table shows two documents, each with the following fields:

- `_id`: ObjectId("52cdef7c4bab8bd675297da8")
- `name`: "OpenX"
- `founded_year`: 2008

The interface also includes a "VIEW" button, a "DISPLAYING DOCUMENTS 1 - 20 OF 1224" status bar, and a "REFRESH" button.

5. Obtén todos los *post* del autor machine.

```
{
  filter: {
    author: 'machine'
  }
}
```

The screenshot shows a MongoDB query interface. The filter is set to `{author: 'machine'}`. The interface includes tabs for FILTER, PROJECT, SORT, and COLLATION. The MAXTIMEMS is set to 5000. The results are displayed as a list of documents. The first document is:

```
{
  "_id": ObjectId("50ab0f8bbcf1bfe2536dc3f9"),
  "body": "Amendment I",
  "permalink": "aRjNnLZkTyspAIoRGe",
  "author": "machine",
  "title": "Bill of Rights",
  "tags": Array,
  "comments": Array,
  "date": 2012-11-20T05:05:15.231+00:00
}
```

The second document is:

```
{
  "_id": ObjectId("50ab0f8bbcf1bfe2536dc3fa"),
  "body": "We the People of the United States, in Order to form a more perfect Un...",
  "permalink": "jNsg0boVWyKEoXNydtis",
  "author": "machine",
  "title": "US Constitution",
  "tags": Array,
  "comments": Array,
  "date": 2012-11-20T05:05:15.232+00:00
}
```

The interface also shows a status bar indicating "Displaying documents 1 - 20 of 500" and a REFRESH button.

6. Obtén todos los tweets provenientes de la web.

```
{
  filter: {
    source: 'web'
  }
}
```

The screenshot shows a MongoDB query interface. The filter is set to `{source: 'web'}`. The interface includes tabs for FILTER, PROJECT, SORT, and COLLATION. The MAXTIMEMS is set to 5000. The results are displayed as a list of documents. The first document is:

```
{
  "_id": ObjectId("5c8ecb0caa187d17ca623f5"),
  "text": "eu preciso de terminar de fazer a minha tabela, está muito foda **",
  "in_reply_to_status_id": null,
  "retweet_count": null,
  "contributors": null,
  "created_at": "Thu Sep 02 18:11:23 +0000 2010",
  "geo": null,
  "source": "web",
  "coordinates": null,
  "in_reply_to_screen_name": null,
  "truncated": false,
  "entities": Object,
  "retweeted": false,
  "place": null,
  "user": Object,
  "favorited": false,
  "in_reply_to_user_id": null,
  "id": 22819396900
}
```

The second document is:

```
{
  "_id": ObjectId("5c8ecb0caa187d17ca623ff"),
  "text": "First week of school is over :P",
  "in_reply_to_status_id": null,
  "retweet_count": null,
  "contributors": null,
  "created_at": "Thu Sep 02 18:11:25 +0000 2010",
  "geo": null,
  "source": "web",
  "coordinates": null,
  "in_reply_to_screen_name": null,
  "truncated": false
}
```

The interface also shows a status bar indicating "Displaying documents 1 - 20 of 11141" and a REFRESH button.

7. Obtén todas las compañías fundadas en octubre del 2008.

```
{
  filter: {
    $and: [
      {
        founded_year: 2008
      },
      {
        founded_month: 10
      }
    ]
  }
}
```

The screenshot shows a MongoDB query interface. The filter bar contains the following query: `{ $and: [ { founded_year: 2008 }, { founded_month: 10 } ] }`. The interface includes buttons for **FIND**, **RESET**, and **OPTIONS**. Below the filter bar, there are tabs for **PROJECT**, **SORT**, and **COLLATION**. The **MAXTIMES** is set to 5000, **SKIP** is 0, and **LIMIT** is 0. The results section shows a table with columns **VIEW**, **JSON**, and **GRID**. The first document is displayed in the viewer, showing fields like `_id`, `name`, `permalink`, `crunchbase_url`, `homepage_url`, `blog_url`, `blog_feed_url`, `twitter_username`, `category_code`, `number_of_employees`, `founded_year`, `founded_month`, `founded_day`, `deadpooled_year`, `deadpooled_month`, `deadpooled_day`, `deadpooled_url`, `tag_list`, `alias_list`, `email_address`, `phone_number`, `description`, `created_at`, `updated_at`, and `overview`. The second document is partially visible, showing `_id` and `name`.

```
{
  _id: ObjectId("52cdef7c4bab8bd6752985ca")
  name: "tunesBag"
  permalink: "tunesbag"
  crunchbase_url: "http://www.crunchbase.com/company/tunesbag"
  homepage_url: "http://www.tunesBag.com"
  blog_url: "http://tunesBag.blogspot.com/"
  blog_feed_url: "http://blog.tunesbag.com/feeds/posts/default?alt=rss"
  twitter_username: ""
  category_code: "games_video"
  number_of_employees: null
  founded_year: 2008
  founded_month: 10
  founded_day: 1
  deadpooled_year: null
  deadpooled_month: null
  deadpooled_day: null
  deadpooled_url: null
  tag_list: "music, cloud, locker, mp3, music-streaming, streaming"
  alias_list: ""
  email_address: "office@tunesBag.com"
  phone_number: "+43 680 215 27 96"
  description: "Social Music Player"
  created_at: "Thu Mar 20 15:45:40 UTC 2008"
  updated_at: "Thu Jan 19 00:37:48 UTC 2012"
  overview: "<p>Austria based tunesBag is a music player with social features on th..."
}
```

```
{
  _id: ObjectId("52cdef7d4bab8bd675298d21")
  name: "Muecs"
}
```

8. Obtén todas las compañías con más de 50 empleados.

```
{
  filter: {
    number_of_employees: {
      $gt: 50
    }
  }
}
```

The screenshot shows the MongoDB Compass interface. At the top, the filter bar contains the query: `{ number_of_employees: { $gt: 50 } }`. Below the filter bar, there are tabs for PROJECT, SORT, and COLLATION. The MAXTIMES is set to 500, SKIP is 0, and LIMIT is 0. The interface shows "Displaying documents 1 - 20 of 793". The first document is for Facebook, with the following fields:

```
{
  "_id": ObjectId("52cdef7c4bab8bd675297d8e"),
  "name": "Facebook",
  "permalink": "facebook",
  "crunchbase_url": "http://www.crunchbase.com/company/facebook",
  "homepage_url": "http://facebook.com",
  "blog_url": "http://blog.facebook.com",
  "blog_feed_url": "http://blog.facebook.com/atom.php",
  "twitter_username": "facebook",
  "category_code": "social",
  "number_of_employees": 5299,
  "founded_year": 2004,
  "founded_month": 2,
  "founded_day": 1,
  "deadpooled_year": null,
  "deadpooled_month": null,
  "deadpooled_day": null,
  "deadpooled_url": "",
  "tag_list": "facebook, college, students, profiles, network, online-communities, so...",
  "alias_list": "",
  "email_address": "",
  "phone_number": "",
  "description": "Social network",
  "created_at": "Fri May 25 21:22:15 UTC 2007",
  "updated_at": "Thu Nov 21 19:40:55 UTC 2013",
  "overview": "<p>Facebook is the world's largest social network, with over <a ...>"
}
```



9. Obtén las historias con número de comentarios entre 10 y 30.

```
{
  filter: {
    $and: [
      {
        comments: {
          $gte: 10
        }
      },
      {
        comments: {
          $lte: 30
        }
      }
    ]
  }
}
```

The screenshot shows a MongoDB query interface. The filter bar contains the query: `{ $and: [{ comments: { $gte: 10 } }, { comments: { $lte: 30 } }] }`. The project bar is empty. The sort bar is empty. The results list shows two documents:

```
{
  "_id": "ObjectId('4ba267dc238d3ba3ca000006')",
  "href": "http://digg.com/travel_places/11_Amazing_Treehouses_from_Around_the_Wo...",
  "title": "11 Amazing Treehouses from Around the World",
  "comments": 15,
  "container": Object,
  "submit_date": 1268772915,
  "topic": Object,
  "promote_date": 1268858405,
  "id": "19970430",
  "media": "news",
  "digg": 375,
  "description": "Treehouses bring us closer to nature, and appeal to the kid in all of ...",
  "link": "http://www.thedailygreen.com/green-homes/latest/treehouse-photos-46031...",
  "user": Object,
  "status": "popular",
  "shorturl": Array
}
```

The second document is:

```
{
  "_id": "ObjectId('4ba267dc238d3ba3ca00000b')",
  "href": "http://digg.com/space/NASA_The_Wizard_Nebula",
  "title": "NASA - The Wizard Nebula ",
  "comments": 14,
  "thumbnail": Object,
  "container": Object,
  "submit_date": 1268745796,
  "topic": Object,
  "promote_date": 1268848202,
  "id": "19960530",
  "media": "images",
  "digg": 434,
  "description": "This image of the open star cluster NGC 7380, also known as the Wizard..."
}
```

10. Obtén la empresa con el menor número de empleados.

```
{
  filter: {
    $and: [
      {
        number_of_employees: {
          $ne: null
        }
      },
      {
        number_of_employees: {
          $ne: 0
        }
      }
    ]
  },
  project: {
    name: 1,
    number_of_employees: 1
  },
  sort: {
    number_of_employees: 1,
    name: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. At the top, there are tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation. The Aggregations tab is active, displaying a pipeline with three stages: FILTER, PROJECT, and SORT. The FILTER stage has a query: `{ $and: [ { number_of_employees: { $ne: null } }, { number_of_employees: { $ne: 0 } } ] }`. The PROJECT stage has a query: `{name:1, number_of_employees:1}`. The SORT stage has a query: `{number_of_employees: 1, name: 1}`. To the right of the SORT stage, there are controls for MAXTIMEMS (5000), SKIP (0), and LIMIT (0). Below the pipeline, there are buttons for FILTER, PROJECT, SORT, and COLLATION. On the right side, there are buttons for FIND, RESET, and a menu icon. Below the pipeline, there are buttons for VIEW, a list icon, a JSON icon, and a table icon. The main area displays the results of the aggregation, showing four documents. Each document has the following structure: `{_id: ObjectId("52cdef7e4bab8bd67529a281"), name: "148Apps", number_of_employees: 1}`. The documents are sorted by `number_of_employees` and then by `name`. The status bar at the bottom indicates "Displaying documents 1 - 20 of 4379" and has a REFRESH button.

_id	name	number_of_employees
ObjectId("52cdef7e4bab8bd67529a281")	148Apps	1
ObjectId("52cdef7d4bab8bd675299706")	360 Sports Software	1
ObjectId("52cdef7d4bab8bd675299a90")	360 Sports Software	1
ObjectId("52cdef7d4bab8bd67529919b")	4chan	1

11. Obtén la empresa con el mayor número de empleados.

```
{
  project: {
    name: 1,
    number_of_employees: 1
  },
  sort: {
    number_of_employees: -1,
    name: 1
  },
  limit: 1
}
```

The screenshot shows a MongoDB query interface. The query is as follows:

```
{
  project: {
    name: 1,
    number_of_employees: 1
  },
  sort: {
    number_of_employees: -1,
    name: 1
  },
  limit: 1
}
```

The interface includes a FILTER section, a PROJECT section, a SORT section, a MAXTIME limit of 5000, a COLLATION section, a SKIP of 0, and a LIMIT of 1. The results show one document:

```
{
  "_id": ObjectId("52cdef7c4bab8bd67529856a"),
  "name": "IBM",
  "number_of_employees": 388000
}
```

12. Obtén la historia más comentada.

```
{
  sort: {
    comments: -1
  },
  limit: 1
}
```

The screenshot shows a MongoDB query interface. The query is as follows:

```
{
  sort: {
    comments: -1
  },
  limit: 1
}
```

The interface includes a FILTER section, a PROJECT section, a SORT section, a MAXTIME limit of 5000, a COLLATION section, a SKIP of 0, and a LIMIT of 1. The results show one document:

```
{
  "_id": ObjectId("4ba27ea0238d3ba3ca002251"),
  "href": "http://digg.com/politics/Republican_Brown_wins_Massachusetts_Senate_se...",
  "title": "Republican Brown wins Massachusetts Senate seat!",
  "comments": 1864,
  "thumbnail": Object,
  "container": Object,
  "submit_date": 1263954202,
  "topic": Object,
  "promote_date": 1263956432,
  "id": "18626778",
  "media": "news",
  "digg": 2098,
  "description": "Striving for an epic upset in liberal Massachusetts, Republican Scott ...",
  "link": "http://news.yahoo.com/s/ap/us_machusetts_senate;_ylt=AuEva3AHQXo0R...",
  "user": Object,
  "status": "popular",
  "shorturl": Array
}
```

13. Obtén la historia menos comentada.

```
{
  filter: {
    comments: {
      $ne: 0
    }
  },
  sort: {
    comments: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. At the top, there are tabs for Documents, Aggregations, Schema, Explain Plan, and Indexes. The Documents tab is active. Below the tabs, there is a query builder section with fields for FILTER, PROJECT, SORT, and COLLATION. The FILTER field contains the query `{comments: {$ne: 0}}`. The SORT field contains `{comments: 1}`. The MAXTIMEMS field is set to 500. The SKIP field is set to 0 and the LIMIT field is set to 0. To the right of the query builder are buttons for FIND, RESET, and a menu icon. Below the query builder, there is a section for ADD, VIEW, and a status bar indicating 'Displaying documents 1 - 20 of 9840'. The VIEW section has buttons for list, JSON, and grid views. The status bar has a REFRESH button. The main area displays two documents in a list view. Each document is a JSON object with fields like `_id`, `href`, `title`, `comments`, `thumbnail`, `container`, `submit_date`, `topic`, `promote_date`, `id`, `media`, `diggs`, `description`, `link`, `user`, `status`, and `shorturl`. The first document has a title 'Smallest eel-loach Fish Discovered' and the second has a title 'Rebtel Buying a Part of Talkster'.

```
{
  "_id": "ObjectId('4ba277c0238d3ba3ca001819')",
  "href": "http://digg.com/general_sciences/Smallest_eel_loach_Fish_Discovered",
  "title": "Smallest eel-loach Fish Discovered",
  "comments": 2,
  "thumbnail": Object,
  "container": Object,
  "submit_date": 1265651817,
  "topic": Object,
  "promote_date": 1265703003,
  "id": "19106141",
  "media": "news",
  "diggs": 176,
  "description": "The world's smallest species of eel-loach fish has been discovered by ...",
  "link": "http://www.physorg.com/news184855470.html",
  "user": Object,
  "status": "popular",
  "shorturl": Array
}
```

```
{
  "_id": "ObjectId('4ba27e1a238d3ba3ca002186')",
  "href": "http://digg.com/tech_news/Rebtel_Buying_a_Part_of_Talkster",
  "title": "Rebtel Buying a Part of Talkster",
  "comments": 2,
  "container": Object,
  "submit_date": 1263997563,
  "topic": Object,
  "promote_date": 1264050002,
  "id": "18638870"
}
```

## PROYECTO SESIÓN 5 CONSULTAS EN MONGODB

1. El proyecto consiste en obtener todas las publicaciones que tengan 50 o más comentarios, que la valoración sea mayor o igual a 80, que cuenten con conexión a Internet vía cable y estén ubicadas en Brazil.

```
{{ $match: {
  number_of_reviews: { $gte: 50 }
}}, { $match: {
  "review_scores.review_scores_rating": { $gte: 80 }
}}, { $match: {
  amenities: { $in: [ /Ethernet/i ] }
}}, { $match: {
  "address.country_code": "BR"
}}, { $group: {
  _id: null,
  total: {
    $sum: 1
  }
}}}
```

The screenshot displays the MongoDB Compass interface with three aggregation pipeline stages. Each stage shows a sample of 20 documents.

**Stage 1: \$match**

Query:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   number_of_reviews: { $gte: 50 }
6 }
```

**Stage 2: \$match**

Query:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "review_scores.review_scores_rating": { $gte: 80 }
6 }
```

**Stage 3: \$match**

Query:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   amenities: { $in: [ /Ethernet/i ] }
6 }
```

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "address.country_code": "BR"
6 }
```

last\_scraped: 2019-02-11T05:00:00.000+00:00  
calendar\_last\_scraped: 2019-02-11T05:00:00.000+00:00  
first\_review: 2014-01-14T05:00:00.000+00:00  
last\_review: 2019-01-02T05:00:00.000+00:00  
accommodates: 4  
bedrooms: 1  
beds: 2

SHOW 14 MORE FIELDS

\_id: "11740322"  
listing\_url: "https://www.airbnb.com/rooms/11740322"  
name: "★Cozy doble bedroom ★Nearby Christ the Redeemer★"  
summary: "Stay in a quiet place with great location! Fr  
wifi access throughout..."  
space: "The house was built in 1927 and is preserved by  
the Municipal Secretar..."  
description: "Stay in a quiet place with great location"

\_id: "14996051"  
listing\_url: "http  
name: "Best locat  
summary: "Perfect  
Farme de  
space: "The bedroo  
very good i  
description: "Per  
Farm

≡

\$group

Output after \$group stage (Sample of 1 document)

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: null,
7   total: {
8     $sum: 1
9   }
10 }
```

\_id: null  
total: 6

## PROYECTO SESIÓN 6 AGREGACIONES

1. El proyecto consiste en obtener, por país, el número de películas que hay de cada género.

```

[{$unwind: {
  path: '$genres'
}}, {$unwind: {
  path: '$countries'
}}, {$group: {
  _id: {
    genero: '$genres',
    pais: '$countries'
  },
  peliculas: {
    $sum: 1
  }
}}, {$addFields: {
  Genero: '$_id.genero',
  Pais: '$_id.pais'
}}, {$project: {
  _id: 0
}}, {$sort: {
  Pais: -1,
  Genero: 1
}}]

```

[illegible]

COLLATION

Untitled

SAVE

SAMPLE MODE

AUTO PREVIEW

Sunwind

Output after \$unwind stage (Sample of 20 documents)

```
1 {
2   path: '$countries'
3 }
```

```
_id: ObjectId("573a1390f29313caabcd4135")
plot: "Three men hammer on an anvil and pass a bottle o
beer around."
genres: "Short"
runtime: 1
cast: Array
num_mflix_comments: 1
title: "Blacksmith scene"
fullplot: "A stationary camera looks at a large anvil w
```

```
_id: ObjectId("573a1390f29313caabcd42e8")
plot: "A group of bandits stage a brazen tr
only to find a determ..."
genres: "Short"
runtime: 11
cast: Array
poster: "https://m.media-
amazon.com/images/H/MV5BMTU3NjE5NzY
title: "The Great Train Robbery"
```

\$group

Output after \$group stage (Sample of 20 documents)

```
1 {
2   _id: {
3     genero: '$genres',
4     pais: '$countries'
5   },
6   peliculas: {
7     $sum: 1
8   }
9 }
```

```
_id: Object
peliculas: 31
```

```
_id: Object
peliculas: 3
```

\$addFields

Output after \$addFields stage (Sample of 20 documents)

```
1 {
2   Genero: '$_id.genero',
3   Pais: '$_id.pais'
4 }
```

```
_id: Object
peliculas: 1
Genero: "Music"
Pais: "Thailand"
```

```
_id: Object
peliculas: 1
Genero: "Animation"
Pais: "Lebanon"
```

\$project

Output after \$project stage (Sample of 20 documents)

```
1 {
2   _id: 0
3 }
```

```
peliculas: 13
Genero: "War"
Pais: "Sweden"
```

```
peliculas: 1
Genero: "Drama"
Pais: "Burma"
```

\$sort

Output after \$sort stage (Sample of 20 documents)

```
1 {
2   Pais: -1,
3   Genero: 1
4 }
```

```
peliculas: 1
Genero: "Drama"
Pais: "Zimbabwe"
```

```
peliculas: 1
Genero: "Comedy"
Pais: "Zaire"
```

## PROYECTO SESIÓN 7 CONFIGURACIÓN DE BASES DE DATOS LOCALES