SQL exercises - Basics and tricks

Marc Herrera

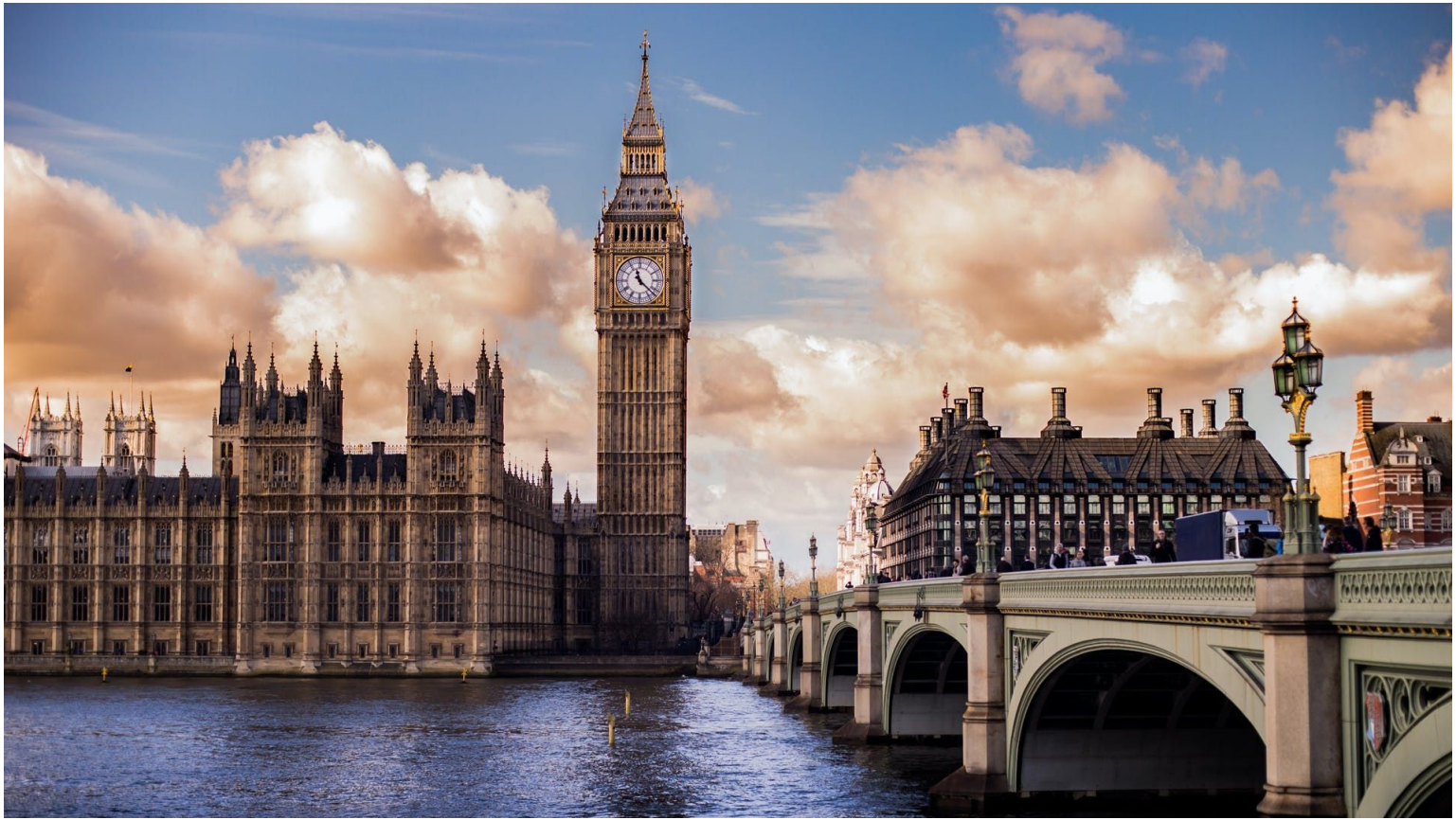
Senior Looker Specialist

Introduction

Database to be used today

- We will use this online console.
- Note that the dialect for this console is SQLite. Some functions may vary (specially date functions).

1; Where are the most of the registered customers from?



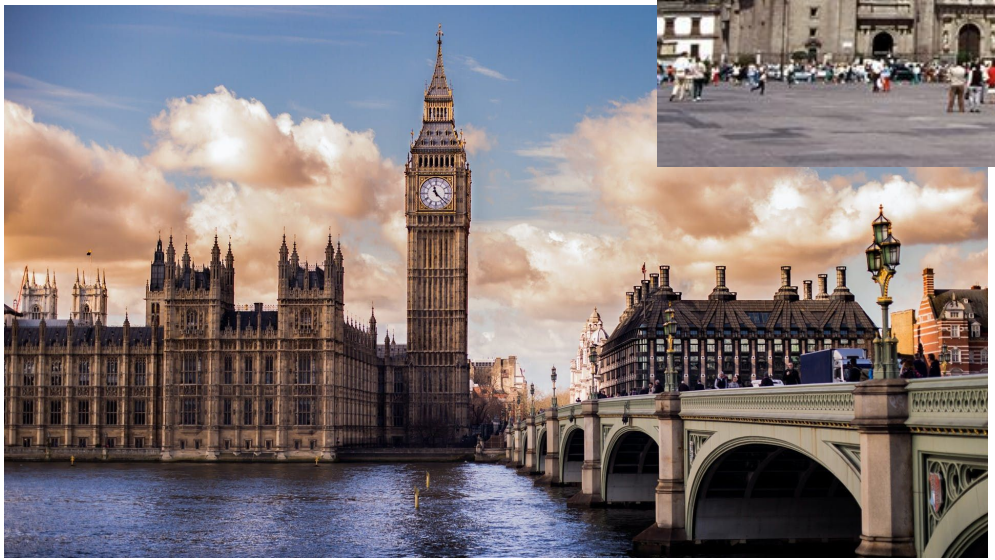
SQL

```
SELECT city, count(*)
FROM Customers
group by 1
ORDER BY 2 desc;
```

Number of Records: 69

City	count(*)
London	6
México D.F.	5
São Paulo	4
Rio de Janeiro	3
Madrid	3
Buenos Aires	3
Portland	2
Paris	2
Nantes	2

2; Where are the most of the customers that ordered at least once from?

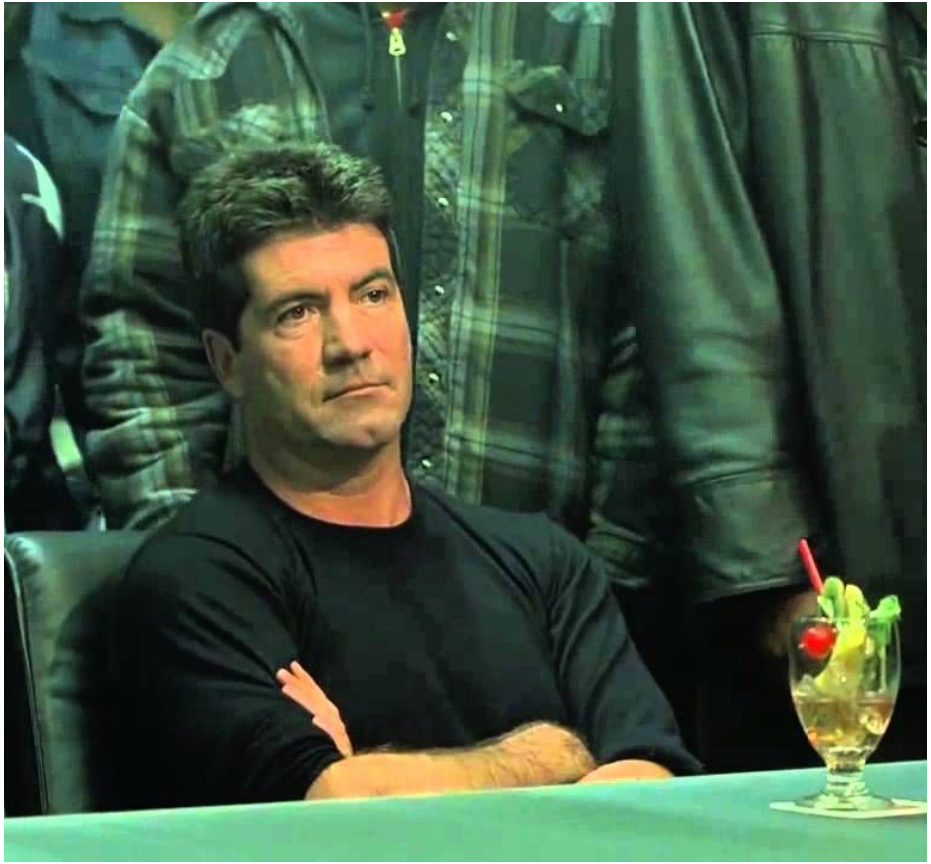


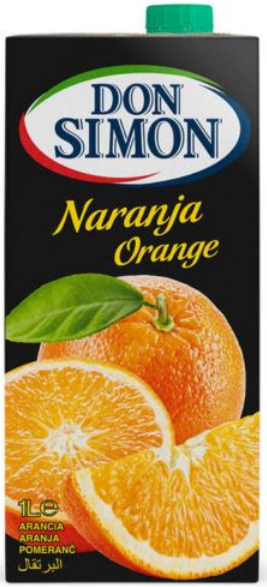
SQL

```
SELECT city, count(distinct customers.customerid)
FROM Customers
join orders on orders.customerid = customers.customerid
group by 1
ORDER BY 2 desc;
```

City	count(distinct
México D.F.	5
London	5
São Paulo	4
Rio de Janeiro	3
Portland	2
Madrid	2
Lisboa	2
Århus	1
Walla	1

3; What’s the name of that London registered customer that has not ordered yet?




DON SIMON
Naranja
Orange
1L
ARANCIA
ARANJA
PORTOCALE
البرتقال

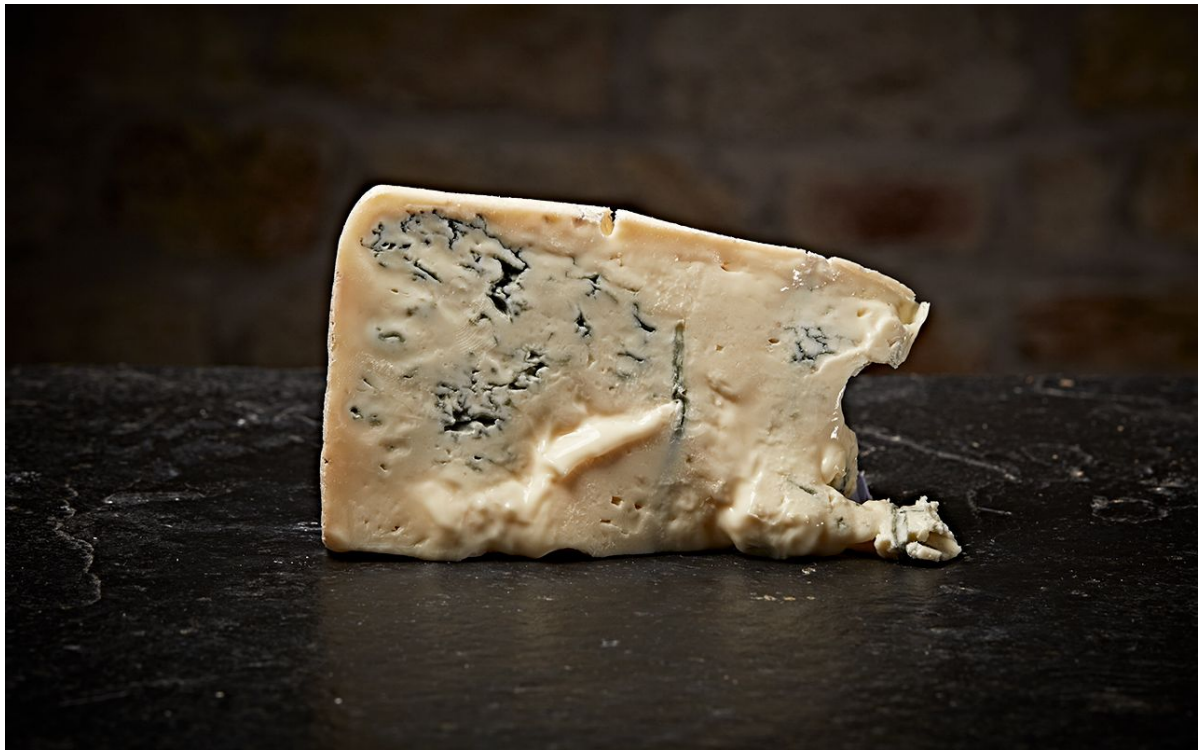

SQL

Number of Records: 1

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
53	North/South	Simon Crowther	South House 300 Queensbridge	London	SW7 1RZ	UK

```
SELECT customers.*
FROM Customers
left join orders on orders.customerid =
customers.customerid
where orders.customerid isnull and city in ('London')
group by 1
ORDER BY 2 desc;
```

4; What’s the most sold product name?

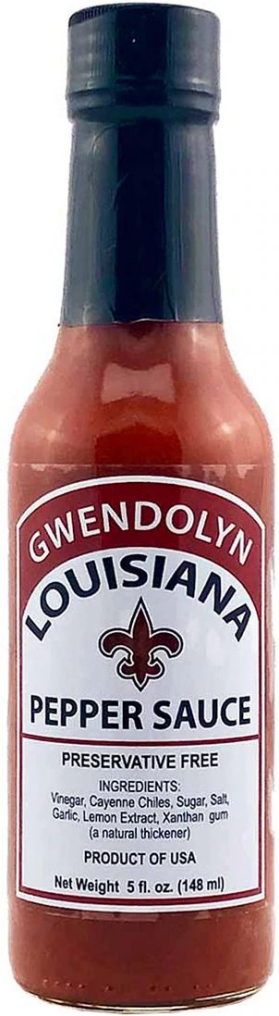


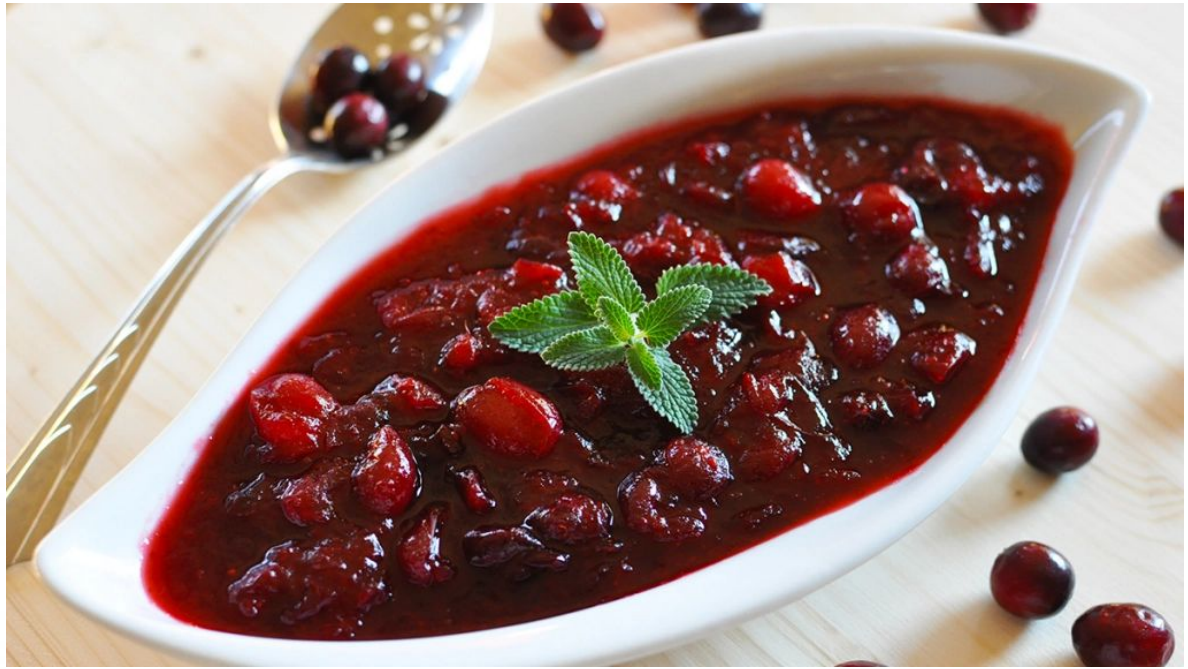

SQL

ProductID	ProductName	sum(quantity)
31	Gorgonzola Telino	458
60	Camembert Pierrot	430
35	Steeleye Stout	369
59	Raclette Courdavault	346
2	Chang	341
16	Pavlova	338

```
SELECT OrderDetails.productID, products.productname,
sum(quantity) FROM OrderDetails
left join products on products.productID =
OrderDetails.productID
group by 1,2 order by 3 desc
```

5; Show product sales taking into account only products that are sauces as per the product name


GWENDOLYN
LOUISIANA
PEPPER SAUCE
PRESERVATIVE FREE
INGREDIENTS:
Vinegar, Cayenne Chiles, Sugar, Salt,
Garlic, Lemon Extract, Xanthan gum
(a natural thickener)
PRODUCT OF USA
Net Weight 5 fl. oz. (148 ml)



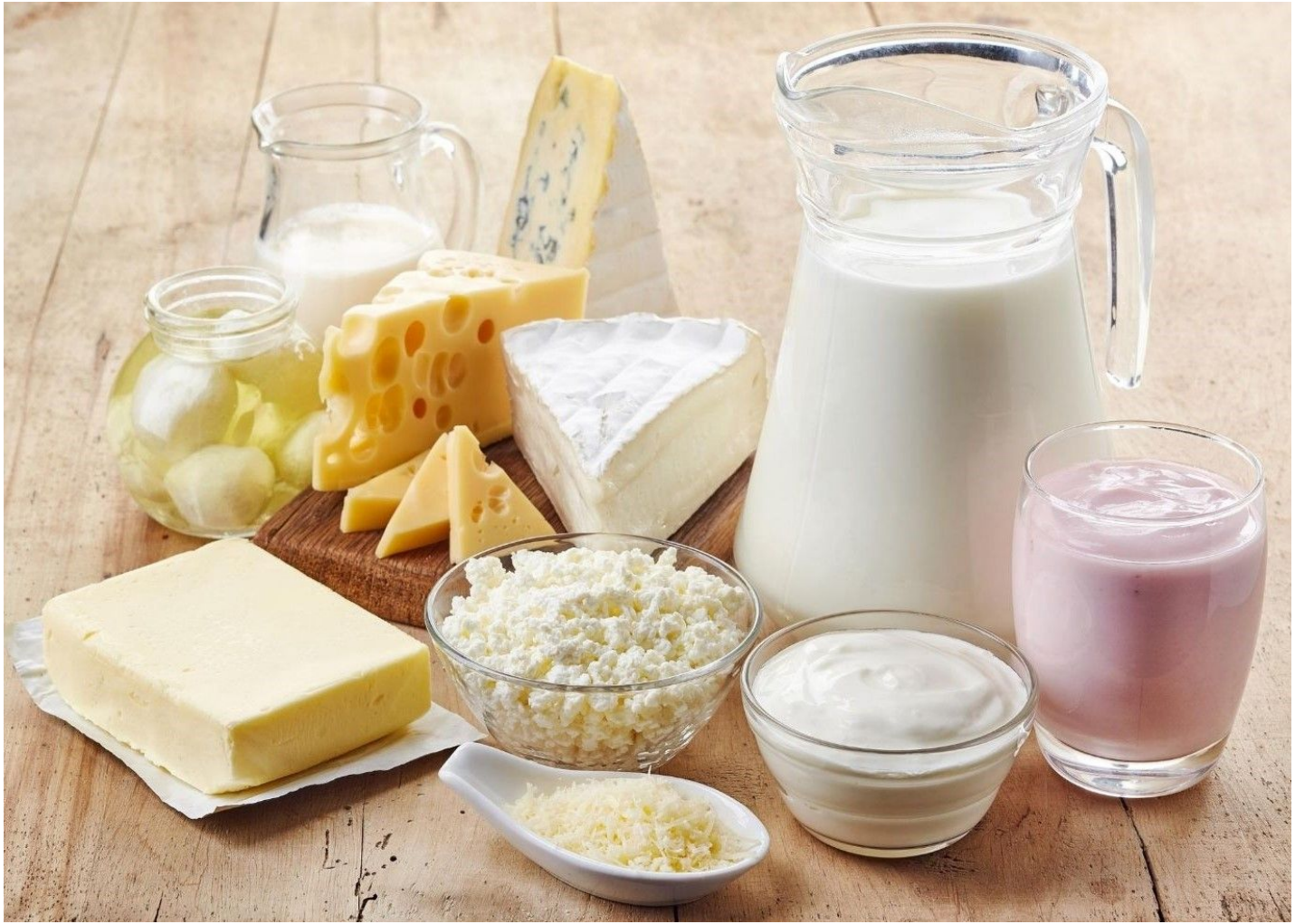
SQL

Number of Records: 2

ProductID	ProductName	sum(quantity)
65	Louisiana Fiery Hot Pepper Sauce	175
8	Northwoods Cranberry Sauce	140

```
SELECT OrderDetails.productID, products.productname,
sum(quantity) FROM OrderDetails
left join products on products.productID =
OrderDetails.productID
where ProductName like '%sauce%'
group by 1,2 order by 3 desc
```

6; What is the most popular category as per number of bought products?



SQL

Number of Records: 8

CategoryName	sum(quantity)
Dairy Products	2601
Beverages	2289
Confections	2110
Seafood	1445
Condiments	1383
Meat/Poultry	1288

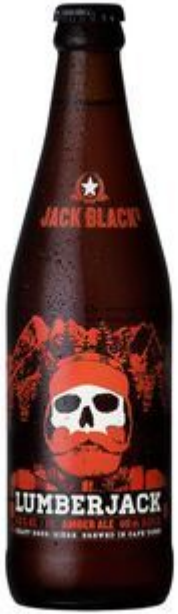
```
SELECT
CategoryName, sum(quantity)
from orderdetails
left join products on orderdetails.productid =
products.productid
left join categories on categories.categoryid =
products.categoryid
group by 1
order by 2 desc
```

Hard;

Who bought the most the least bought product?

Hard; Who bought the most the least bought product?

7; What is the least bought product? Get the result in one line only


JACK BLACK
LUMBERJACK


SQL

ProductName	sum(quantity)
Laughing Lumberjack Lager	5

```
SELECT products.productname, sum(quantity) FROM
OrderDetails
left join products on products.productID =
OrderDetails.productID
group by 1 order by 2 asc
limit 1
```

Hard; Who bought the most **the least bought product**?

8; How would you turn this result into a one column result with only the product name?

That way we will be able to add it in a where clause afterwards

ProductName
Laughing Lumberjack Lager

SELECT productname from

(SELECT products.productname, sum(quantity) FROM

OrderDetails

left join products on products.productID =

OrderDetails.productID

group by 1 order by 2 asc

limit 1**)**

Hard;

9; Who bought the most the least bought product?

LEHMANNs

aus Leidenschaft zum Essen

CustomerName	sum(quantity)
Lehmanns Marktstand	5

```
SELECT customers.customername, sum(quantity)
FROM OrderDetails
left join products on products.productID = OrderDetails.productID
left join orders on orders.orderid = OrderDetails.orderid
left join customers on customers.customerid = orders.customerid
where products.productname in(
SELECT productname from
(SELECT products.productname, sum(quantity) FROM OrderDetails
left join products on products.productID = OrderDetails.productID
group by 1 order by 2 asc
limit 1))
group by 1
order by 2 desc
```

Hard;

10; building up a retention cohort

Hint; you need time difference functions and subqueries to build this

Retention cohort

We will be building a customer retention cohort . I recommend you to to do it before we have the class. We will be correcting it then. It is the most asked question across interviews .

- Online console https://www.w3schools.com/sql/trysql.asp?filename=trysql_asc
- We want to see **how retention evolved across months for each of the bucket customers**. In the rows we can see the distinct customers that ordered based on when they first ordered. The columns show how many months it passed since they first ordered.

(Once you get the needed information you can do the pivot outside by just copy pasting the table into an excel file, no worries about the visualization part)

First step; figure what information we do need

SUM of count(*)	Cohortmonth							
FirstOrderMonth	0	1	2	3	4	5	6	7
1996-07	100%	15%	25%	25%	20%	35%	30%	15%
1996-08	100%	50%	21%	29%	29%	21%	21%	
1996-9	100%	0%	25%	13%	0%	13%		
1996-10	100%	33%	22%	44%	22%			
1996-11	100%	80%	30%	10%				
1996-12	100%	60%						
1997-01	100%							
1997-02	100%							

Second step; realising what I need to get that. How can I get the first order date per customer?


```
SELECT
        CustomerID,
        min(OrderDate) as firstorderdoneon
FROM orders
group by 1
```

Number of Records: 74

CustomerID	firstorderdoneon
2	1996-09-18
3	1996-11-27
4	1996-11-15
5	1996-08-12
7	1996-07-25
8	1996-10-10
9	1996-10-16
10	1996-12-20
11	1996-08-26
13	1996-07-18
14	1996-07-11
15	1996-08-27
16	1997-02-04
17	1996-11-26
18	1996-09-30

Third step; join that new information using a subquery and format as a month


```
with firstorderpercustomer as (SELECT CustomerID, min(OrderDate) as firstorderdoneon
FROM orders group by 1 )
SELECT
        strftime('%Y-%m', firstorderdoneon) as FirstOrderMonth,
        firstorderdoneon,
        OrderDate,
        julianday(firstorderdoneon) julianFirstOrderDate,
        julianday(OrderDate) JulianOrderDate,
        (julianday(OrderDate) -  julianday(firstorderdoneon)) CohortDay,
        round((julianday(OrderDate) -  julianday(firstorderdoneon))/30) as cohortmonth
from Orders
left join firstorderpercustomer on firstorderpercustomer.CustomerID =
orders.CustomerID
order by 3 desc
```

Third step; join that new information using a subquery and format as a month

Number of Records: 196

FirstOrderMonth	firstorderdoneon	OrderDate	julianFirstOrderDate	JulianOrderDate	CohortDay	cohortmonth
1996-08	1996-08-23	1997-02-12	2450318.5	2450491.5	173	6
1996-07	1996-07-17	1997-02-11	2450281.5	2450490.5	209	7
1996-10	1996-10-08	1997-02-10	2450364.5	2450489.5	125	4
1996-07	1996-07-19	1997-02-10	2450283.5	2450489.5	206	7
1996-10	1996-10-17	1997-02-07	2450373.5	2450486.5	113	4
1997-02	1997-02-06	1997-02-06	2450485.5	2450485.5	0	0
1996-07	1996-07-25	1997-02-05	2450289.5	2450484.5	195	7
1996-07	1996-07-26	1997-02-05	2450290.5	2450484.5	194	6
1997-02	1997-02-04	1997-02-04	2450483.5	2450483.5	0	0
1996-10	1996-10-23	1997-02-03	2450379.5	2450482.5	103	3
1996-07	1996-07-24	1997-02-03	2450288.5	2450482.5	194	6
1996-08	1996-08-01	1997-01-31	2450296.5	2450479.5	183	6
1996-07	1996-07-17	1997-01-30	2450281.5	2450478.5	197	7
1996-12	1996-12-20	1997-01-30	2450437.5	2450478.5	41	1
1996-09	1996-09-05	1997-01-29	2450331.5	2450477.5	146	5
1996-08	1996-08-23	1997-01-28	2450318.5	2450476.5	158	5
1996-11	1996-11-28	1997-01-27	2450415.5	2450475.5	60	2

Fourth step; group the orders based on the month and cohort

with firstorderpercustomer as (SELECT CustomerID, min(OrderDate) as firstorderdoneon FROM orders group by 1)
SELECT
strftime('%Y-%m', firstorderdoneon) as ***FirstOrderMonth****,*
round((julianday(OrderDate) - julianday(firstorderdoneon))/30) as ***cohortmonth***
from orders
left join firstorderpercustomer on firstorderpercustomer.CustomerID = orders.CustomerID
group by 1,2

Number of Records: 32

FirstOrderMonth	cohortmonth
1996-07	0
1996-07	1
1996-07	2
1996-07	3
1996-07	4
1996-07	5
1996-07	6
1996-07	7
1996-08	0
1996-08	1
1996-08	2
1996-08	3
1996-08	4

Fifth step; decide aggregation calculation

Solution code;

with firstorderpercustomer as (SELECT CustomerID, min(OrderDate) as firstorderdoneon FROM orders group by 1)
SELECT
strftime('%Y-%m', firstorderdoneon) as FirstOrderMonth,
round((julianday(OrderDate) - julianday(firstorderdoneon))/30) as cohortmonth,
count(distinct orders.CustomerID) as distinctcust
from orders
left join firstorderpercustomer on firstorderpercustomer.CustomerID = orders.CustomerID
group by 1,2
order by 3 desc

Final step; visualize

Number of Records: 32

FirstOrderMonth	cohortmonth	distinctcust
1996-07	0	20
1996-08	0	14
1996-11	0	10
1996-10	0	9
1996-09	0	8
1996-11	1	8
1996-07	5	7
1996-08	1	7
1996-07	6	6
1997-01	0	6

SUM of distinctc cohortmonth								
FirstOrderMonth	0	1	2	3	4	5	6	7
1996-07	20	3	5	5	4	7	6	3
1996-08	14	7	3	4	4	3	3	
1996-9	8		2	1		1		
1996-10	9	3	2	4	2			
1996-11	10	8	3	1				
1996-12	5	3						
1997-01	6							
1997-2	2							

SUM of count(*)	Cohortmonth							
FirstOrderMonth	0	1	2	3	4	5	6	7
1996-07	100%	15%	25%	25%	20%	35%	30%	15%
1996-08	100%	50%	21%	29%	29%	21%	21%	
1996-9	100%	0%	25%	13%	0%	13%		
1996-10	100%	33%	22%	44%	22%			
1996-11	100%	80%	30%	10%				
1996-12	100%	60%						
1997-01	100%							
1997-02	100%							

Thanks =)