

Please check my implementation of all exercises from the following captures.

Exercise: Create a virtual table to summarize data

```
-- Week2
4  -- Task1
5  CREATE VIEW OrdersView AS SELECT OrderID,Quantity,Cost FROM Orders WHERE Quantity > 2;
6  SELECT * FROM OrdersView;
```

100% 9:8

Result Grid Filter Rows: Search Export:

OrderID	Quantity	Cost
00-465-7031	3	320.00
00-618-9205	3	124.62
00-799-3091	3	101.60
00-822-8333	3	220.00
01-423-8434	3	320.00
01-715-3869	3	52.08
01-855-8188	3	320.00
01-879-8004	3	98.00
01-975-7149	3	210.00
02-211-2785	3	125.00
02-715-2965	3	320.00
02-732-6800	3	310.00
02-739-5815	3	75.00

OrdersView 7

Query 1 Exercises

Limit to 1000 rows

```
5  CREATE VIEW OrdersView AS SELECT OrderID,Quantity,Cost FROM Orders WHERE Quantity > 2;
6  SELECT * FROM OrdersView;
7
8  -- Task2
9  SELECT Orders.CustomerID,Customers.CustomerName,Orders.OrderID,Orders.Cost,
10 MenuItems.StarterName AS MenuName, MenuItems.CourseName
11 FROM Orders,Customers,MenuItems
12 WHERE Orders.Cost > 150
13 ORDER BY CustomerID
14 DESC;
```

100% 20:13

Result Grid Filter Rows: Search Export: Fetch rows:

CustomerID	CustomerName	OrderID	Cost	MenuName	CourseName
91-003-5808	Tatiana O'Crigane	15-192-4862	220.00	Olives	Greek salad
62-555-9031	Tatiana O'Crigane	15-260-9895	190.00	Olives	Greek salad
09-488-5773	Tatiana O'Crigane	15-307-8168	220.00	Olives	Greek salad
21-282-1343	Tatiana O'Crigane	15-663-0481	235.00	Olives	Greek salad
33-882-5916	Tatiana O'Crigane	16-386-7757	225.00	Olives	Greek salad
32-572-1979	Tatiana O'Crigane	16-946-9856	225.00	Olives	Greek salad
03-575-7094	Tatiana O'Crigane	17-146-9681	310.00	Olives	Greek salad
99-534-0752	Tatiana O'Crigane	17-396-0834	190.00	Olives	Greek salad
98-778-3557	Tatiana O'Crigane	17-708-5350	225.00	Olives	Greek salad
67-683-3934	Tatiana O'Crigane	18-252-1922	210.00	Olives	Greek salad
59-386-1239	Tatiana O'Crigane	19-329-8137	190.00	Olives	Greek salad

Result 2

```

15
16  -- Task3
17  SELECT MenuItem.StarterName AS MenuName FROM MenuItem
18  WHERE EXISTS (SELECT Orders.Quantity FROM Orders WHERE Orders.Quantity > 2);
19
20

```

100% 77:18

Result Grid Filter Rows: Search Export: Fetch rows:

	MenuName
	Olives
	Flatbread
	Minestrone
	Tomato bread
	Falafel
	Hummus
	Olives
	Flatbread
	Minestrone
	Tomato bread

Exercise: Create optimized queries to manage and analyze data

```

20  -- Creating stored procedure
21  -- Task1
22  DELIMITER //
23
24  CREATE PROCEDURE GetMaxQuantity()
25  BEGIN
26  SELECT MAX(Quantity) as MaxQuantity FROM Orders;
27  END //
28
29  DELIMITER ;
30  CALL GetMaxQuantity();
31

```

100% 23:30

Result Grid Filter Rows: Search Export:

	MaxQuantity
3	

```
31
32 -- Task2 Creating prepared statement
33 • PREPARE GetOrderDetail FROM "SELECT OrderID,Quantity,Cost FROM Orders WHERE OrderID = ?";
34 • SET @id = '03-132-2890';
35 • Execute GetOrderDetail using @id;
36
```

100% 34:35

Result Grid Filter Rows: Search Export:

OrderID	Quantity	Cost
03-132-2890	2	75.00

Limit to 1000 rows

```
38 DELIMITER //
39
40 • CREATE PROCEDURE CancelOrder(IN order_id VARCHAR(255))
41 • BEGIN
42 • SET @order_id_out = order_id;
43 • DELETE FROM Orders WHERE OrderID = order_id;
44 • INSERT INTO OrderStatus (Confirmation) VALUES (CONCAT("Order ",@order_id_out," is cancelled"));
45 • END //
46
47 DELIMITER ;
48 • CALL CancelOrder("00-669-6040");
49
50 • SELECT * FROM OrderStatus;
51 •
```

100% 7:45

Result Grid Filter Rows: Search Edit: Export/Import:

ID	Confirmation
1	Order 00-669-6040 is cancelled
NULL	NULL

Exercise: Create SQL queries to check available bookings based on user input

```
1 CREATE TABLE IF NOT EXISTS Bookings (  
2     BookingID INT PRIMARY KEY,  
3     BookingDate DATE,  
4     TableNumber INT,  
5     CustomerID INT);  
6  
7 INSERT INTO Bookings (BookingID,BookingDate, TableNumber,CustomerID)  
8 VALUES(1, '2022-10-10',5,1),  
9         (2, '2022-11-12',3,3),  
10        (3, '2022-10-11',2,2),  
11        (4, '2022-10-13',2,1);  
12  
13 SELECT * FROM Bookings;
```

100% 24:13

Result Grid Filter Rows: Search Edit: Export/Import:

	BookingID	BookingDate	TableNumber	CustomerID
	1	2022-10-10	5	1
	2	2022-11-12	3	3
	3	2022-10-11	2	2

Limit to 1000 rows

```
39  
40 DELIMITER //  
41  
42 CREATE PROCEDURE CheckBooking(IN booking_date DATE, IN table_number INT)  
43 BEGIN  
44     SELECT BookingStatus FROM BookingStatus WHERE BookingDate = booking_date AND TableNumber = table_number;  
45 END //  
46  
47 DELIMITER ;  
48 CALL CheckBooking("2022-11-12",3);  
49  
50 -- DROP PROCEDURE CheckBooking;
```

100% 35:48

Result Grid Filter Rows: Search Export:

BookingStatus
Table 3 is already booked

```

55 CREATE PROCEDURE AddValidBooking(IN booking_date DATE, IN table_number INT)
56 BEGIN
57     SET @b_date = booking_date;
58     SET @t_number = table_number;
59     START TRANSACTION;
60     INSERT INTO Bookings(BookingID,BookingDate,TableNumber,CustomerID) VALUES(6,@b_date,@t_number,5);
61     IF EXISTS(SELECT * FROM Bookings WHERE BookingDate = @b_date AND TableNumber = @t_number) THEN
62         ROLLBACK;
63         INSERT INTO BookingStatus(BookingDate,TableNumber,BookingStatus)
64         VALUES(@b_date,@t_number,CONCAT("Table ",TableNumber," is already booked - booking cancelled"));
65     ELSE
66         COMMIT;
67     END IF;
68 END //
69 DELIMITED ;

```

100% 35:73

Result Grid Filter Rows: Search Export:

BookingStatus
Table 6 is already booked
Table 6 is already booked - booking cancelled

LittleLemon Data Analysis with Tableau

Exercise: Set up the Tableau Workspace for data analysis

Task 1

Tableau - Book1

LittleLemon_data

Connection
☒ Live ☐ Extract

Connections
LittleLemon_data
Text file

Files
☐ Use Data Interpreter
Data Interpreter might be able to clean your Text file workbook.
LittleLemon_data.csv

New Union
New Table Extension

LittleLemon_data.csv

Need more data?
Drag tables here to relate them. [Learn more](#)

LittleLemon_data.csv 21 fields 651 rows 100 rows

#	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv
Row Number	Order ID	Order Date	Delivery Date	Customer ID	Customer Name	City	Country	
137	69-607-3610	21/8/2022 AD	30/3/2021 AD	37-013-0292	Sterne Salterne	Shawnee Mission	United States	
141	56-076-6137	13/8/2020 AD	7/10/2021 AD	46-957-2549	Mady McMennum	Provo	United States	
195	06-887-6332	27/2/2022 AD	19/12/2019 AD	52-934-7220	Merwin Pikhno	Reno	United States	
255	04-904-4552	14/7/2021 AD	23/10/2022 AD	93-077-9628	Aime Ferry	Sacramento	United States	
262	60-081-1759	27/7/2019 AD	20/4/2022 AD	81-155-1642	Merlina Lermmit	Allentown	United States	
348	12-718-1501	5/6/2020 AD	22/10/2019 AD	44-371-9315	Schuyler Walewski	Inglewood	United States	
349	99-748-0471	5/11/2019 AD	4/3/2021 AD	16-794-2589	Ariadne Copeland	Phoenix	United States	
350	84-864-3249	18/12/2019 AD	23/3/2021 AD	75-496-8007	Tamma Clink	Charlotte	United States	
397	10-370-7784	9/12/2019 AD	15/11/2020 AD	01-169-3967	Susette O'Neil	Hamilton	United States	
521	87-435-5817	26/4/2021 AD	1/3/2021 AD	30-481-3036	Aldrich Phythien	Lubbock	United States	
548	61-509-9344	13/2/2021 AD	17/12/2020 AD	23-245-6549	Dusty Yedall	Wilkes Barre	United States	

Task 2

LittleLemon_data

Connection
☒ Live ☐ Extract

Fi
1

LittleLemon_data.csv

Need more data?
Drag tables here to relate them. [Learn more](#)

LittleLemon_data.csv 23 fields 651 rows 100 rows

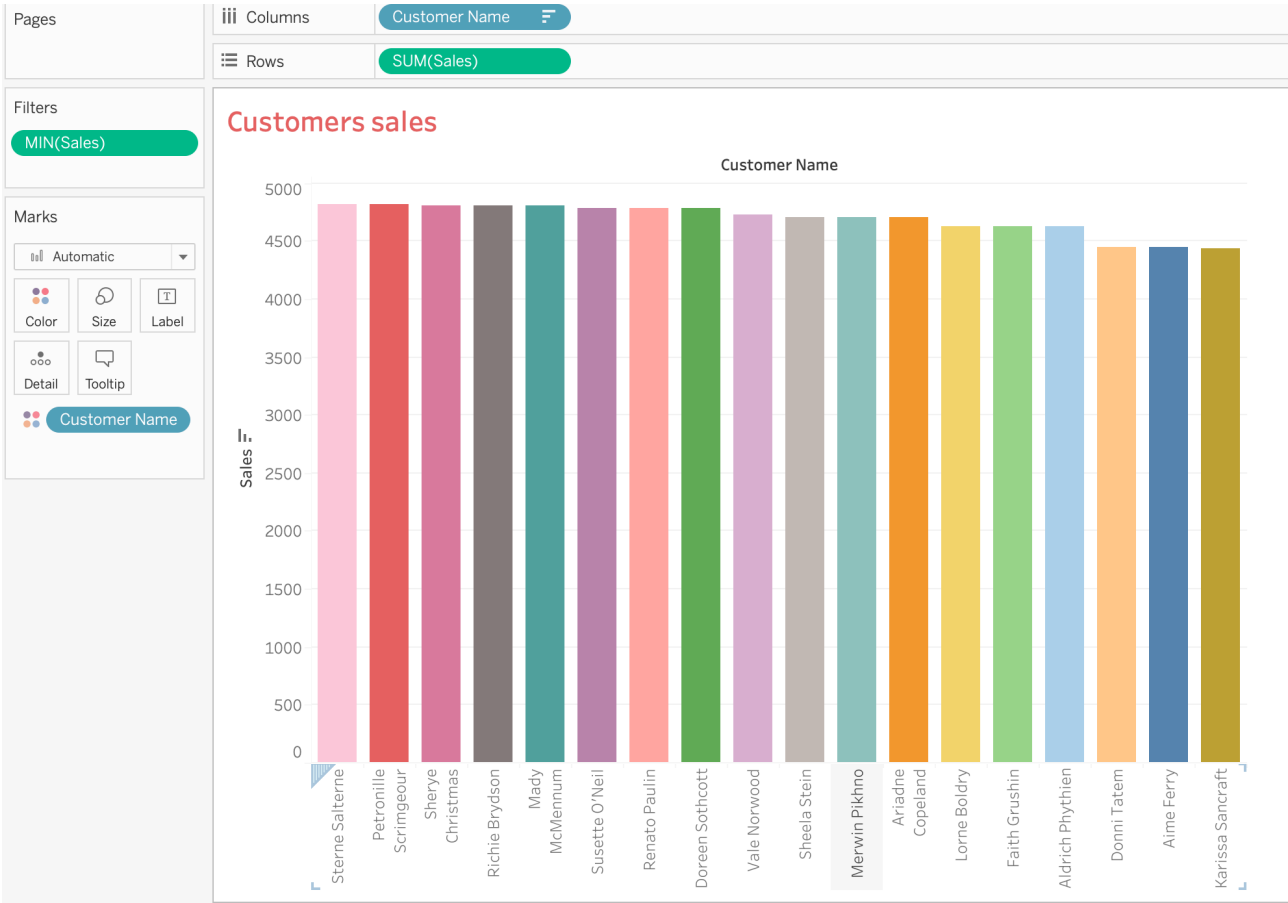
LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	LittleLemon_data.csv	Calculation	Calculation
Course Name	Cuisine Name	Starter Name	Desert Name	Drink	Sides	First Name	Last Name
asa	Greek	Tomato bread	Turkish yoghurt	Ankara White Wine	Meatballs	Sterne	Salterne
za	Greek	Flatbread	Cheesecake	Italian Coffee	Bruschetta	Mady	McMennum
za	Italian	Falafel	Cheesecake	Italian Coffee	Bruschetta	Merwin	Pikhno
za	Italian	Flatbread	Cheesecake	Italian Coffee	Bruschetta	Aime	Ferry
bonara	Italian	Minestrone	Affogato	Roma Red wine	Focaccia	Merlina	Lermmit
varma	Turkish	Hummus	Baklava	Turkish Coffee	Fries	Schuyler	Walewski
ek salad	Greek	Olives	Greek yoghurt	Athens White wine	Tapas	Ariadne	Copeland
n soup	Italian	Flatbread	Ice cream	Corfu Red Wine	Potato salad	Tamma	Clink
ek salad	Greek	Falafel	Greek yoghurt	Athens White wine	Tapas	Susette	O'Neil
asa	Greek	Flatbread	Turkish yoghurt	Ankara White Wine	Meatballs	Aldrich	Phythien
n soup	Turkish	Tomato bread	Ice cream	Corfu Red Wine	Potato salad	Dusty	Yedall

Task 3

a.csv	# LittleLemon_data.c...	# LittleLemon_data.c...	=# Calculation	
e	Cost	Sales	Profit	
	190.000	285.000	95.000	
	144.090	216.135	72.045	
	125.000	187.500	62.500	
	91.840	137.760	45.920	
	235.000	352.500	117.500	
	119.800	179.700	59.900	
	125.000	187.500	62.500	
	235.000	352.500	117.500	
	320.000	480.000	160.000	
	52.080	78.120	26.040	
	235.000	352.500	117.500	

Exercise: Create interactive dashboard for sales and profits

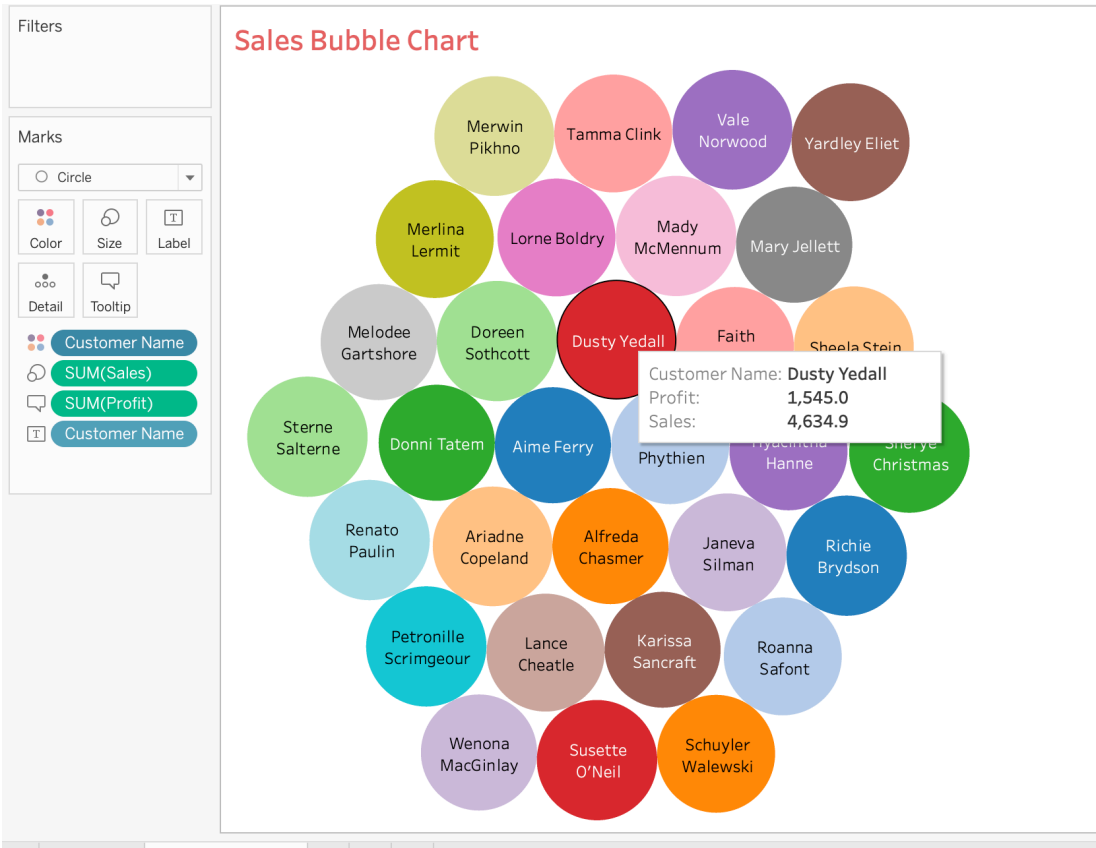
Task 1



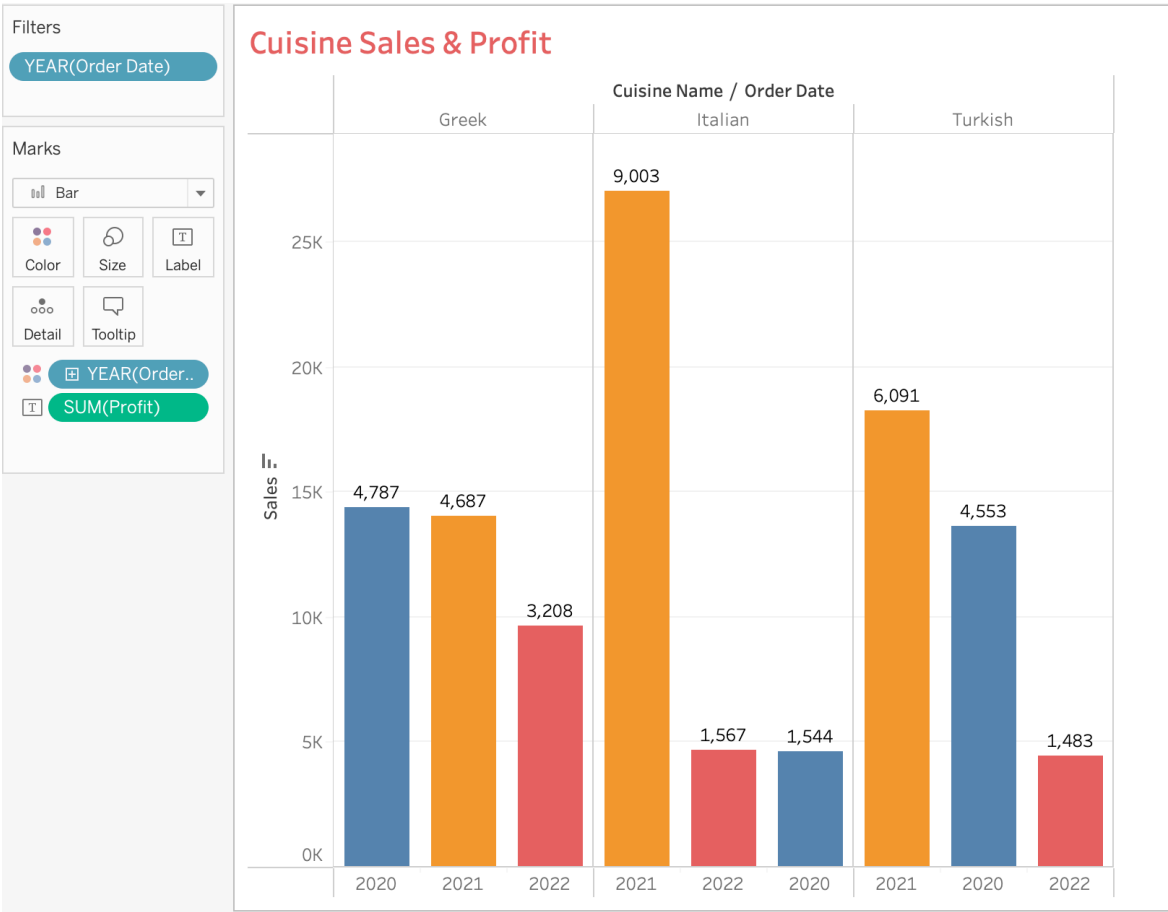
Task 2



Task 3

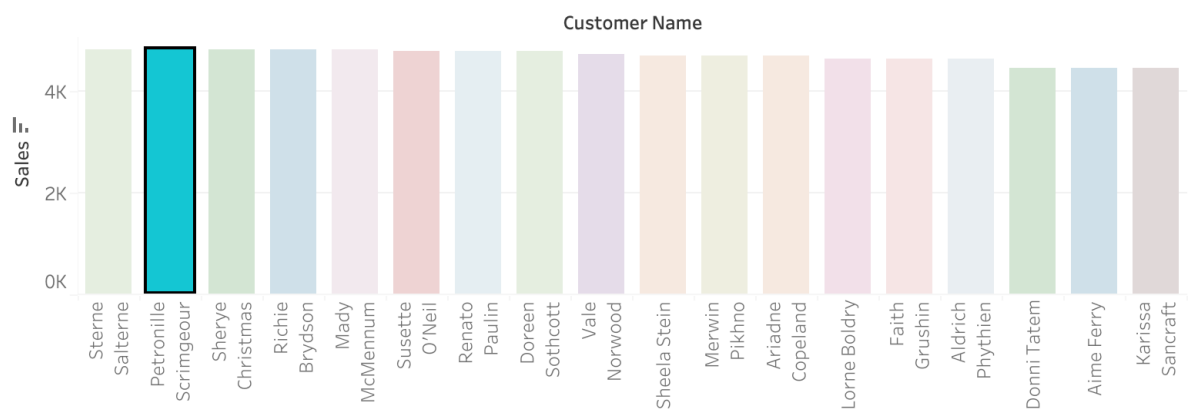


Task 4

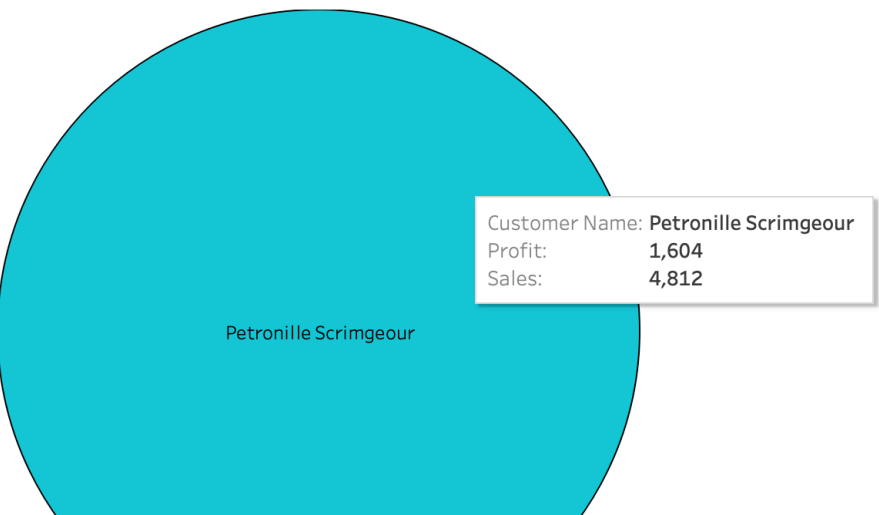


Task 5

Customers sales



Sales Bubble Chart



Exercise: Set up the client project

Task 1

```
> python --version  
Python 3.10.11
```

~/project/metaDatabase

Task 2

Jupyter notebook installed.

Task 3

mysql-connector-python installed.

Exercise: Add query functions

Task 1 & Task 2

```
In [7]: # Import mysql connector and connect to the data base  
import mysql.connector as connector  
  
connection=connector.connect(user="root",password="",db = "LittleLemonDB")  
cursor=connection.cursor()
```

```
In [8]: show_tables_query = "SHOW tables"  
cursor.execute(show_tables_query)  
results = cursor.fetchall()
```

```
In [9]: print(results)  
  
[('Bookings',), ('BookingStatus',), ('Customers',), ('MenuItems',), ('Menus',), ('Orders',), ('OrderStatus',), ('ordersview',)]
```

```
In [ ]:
```

Task 3

```
In [10]: query = '''SELECT Customers.CustomerName, Orders.Sales FROM Orders
            INNER JOIN Customers
            ON Customers.CustomerID = Orders.CustomerID
            WHERE Orders.Sales > 60'''
```

```
In [11]: cursor.execute(query)
results = cursor.fetchall()
```

```
In [14]: print(cursor.column_names)
for i in range(10): #printout only first 10 results
    print(results[i])

('CustomerName', 'Sales')
('Pansie Alldis', Decimal('252.41'))
('Xymenes Hands', Decimal('139.70'))
('Lulu Hastin', Decimal('352.50'))
('Salvidor Francesch', Decimal('315.00'))
('Noni Quickenden', Decimal('480.00'))
('Jilleen Foux', Decimal('186.93'))
('Grenville Hyne', Decimal('216.14'))
('Ram Sinclair', Decimal('152.40'))
('Jilly Bursnoll', Decimal('330.00'))
('Gabe Boyan', Decimal('139.70'))
```

```
In [1]:
```