SQL Assignment

PART-1

Create the schema given below and solve the following questions.
You can use http://sqlfiddle.com/ to build the schema, i.e., create table and insert values (Select PostgreSQL 9.6 as database)

```
CREATE TABLE customer_order(order_num int, cust_id int , order_date date);

INSERT INTO customer_order VALUES
(1,121,'2019-01-15'),(2,234,'2019-07-24'),(3,336,'2020-05-02'),
(4,121,'2019-01-15'),(5,336,'2020-03-19'),(6,234,'2019-07-24'),(7,121,'2019-01-05'),(8,3
36,'2020-06-12');

CREATE TABLE customer (cust_id int, cust_name varchar(40));

INSERT INTO customer VALUES (121, 'Acme Wholesalers'),(234, 'Griifin Electronics'),
(336, 'East Coast Marine Supplies'), (544, 'Sanford Automotive');
```

1. Along with the customer_order table, there is another customer table below. Write a query that returns the name of each customer who has placed exactly 3 orders. Do not return the same customer name more than once, and use a correlated subquery (no JOINS please) against Customer_Order to determine the total number of orders for each customer.

2. Construct a different query to return the same data as the previous question (name of each customer who has placed exactly 3 orders) but use a non-correlated subquery (no JOINS please) against the Customer_Order table. It is important to code a non-correlated subquery for this question.

PART-2

Create the schema given below and solve the following questions.
You can use http://sqlfiddle.com/ to build the schema, i.e., create table and insert values (Select PostgreSQL 9.6 as database)

```
CREATE TABLE region (region_id Integer not null, region_name varchar(50),
super_region_id integer,
primary key(region_id), foreign key (super_region_id) references region(region_id)) ;

INSERT INTO region VALUES (101,'North America',null)
,(102,'USA',101),(103,'Canada',101),(104,'USA-Northeast',102),
(105,'USA-Southeast',102),(106,'USA-West',102),(107,'Mexico',101);

CREATE TABLE product (product_id integer,product_name varchar(50), primary
key(product_id));

INSERT INTO product VALUES (1256,'Gear-Large'),(4437,'Gear
Small'),(5567,'Crankshaft'),(7684,'Sprocket');

CREATE TABLE sales_totals (product_id integer,region_id integer, year integer,month
integer, sales integer,
primary key(product_id,region_id,year,month), foreign key (product_id) references
product(product_id),
foreign key (region_id) references region(region_id));

INSERT INTO sales_totals VALUES
(1256,104,2020,1,1000),(4437,105,2020,2,1200),(7684,106,2020,3,800),(1256,103,2
020,4,2200),(4437,107,2020,5,1700),(7684,104,2020,6,750),(1256,104,2020,7,1100),
(4437,105,2020,8,1050),(7684,106,2020,9,600),(1256,103,2020,10,1900),(4437,107,
2020,11,1500),(7684,104,2020,12,900);
```

3. Write a query that will pivot the sales_totals data so that there is a column for each of the 4 products containing the total sales across all months of 2020. It is OK to include the product_id values in your query, and the results should look as follows:

tot_sales_large_gears	tot_sales_small_gears	tot_sales_crankshafts	tot_sales_sprockets
6200	5450	0	3050

- (Optional Bonus Question)
  Write a statement to create a view called product_sales_totals which will group sales data by product and year. Columns should include product_id, year, product_sales, and gear_sales, which will contain the total sales for the “Gear - Large” and “Gear Small” products (should be generated by an expression, and it is OK to use the product_id values in the expression). To accomplish this, you need a CASE statement. The product_sales column should be a sum of sales for the particular product_id and year, regardless of what kind of product it is. The gear_sales column should be a sum of sales only in the case where the product is either "Gear - Large” or “Gear Small”. Else in the case that the product is neither “Gear - Large” or “Gear Small”, the value for gear_sales should be 0.