Analytical SQL Case Study (question 1):

The Main Scenario

- here is **the scenario**, we are covering some countries and we want to know **which countries** we need to focus on by **make campaigns,** support their stocks to be able to handle the customer's orders to achieve the customer satisfaction and which countries we should **ignore**. depending on the **amount of money** we get from those countries.

First

- let's get into **the details**, I will focus only on the **highest and lowest** countries in the **total purchasing**. First i will get the last 5 countries in the total purchasing, total ordering and the **time** between the first and last order to make sure that those countries are **actually inactive**.

Second

- Second i will divide the countries into 5 level depending on the total purchasing they did, then i will get max, min number of orders every country of them made, and which stock who had the max number of orders from them and finally i will find the **most active customers** in those countries to give a **prize** as a kind of marketing.

First Query

The Query

Result

- **first we will check the number of countries we covering:**

```
SELECT count(distinct "Country") as "number of countries"
FROM public."Online_Retail";
```

- **to find the appropriate countries to ignore, i decided to join the lowest 5 countries in total purchasing with the lowest 5 countries in total ordering .**

```
select p."Country", total_Purchases, row_number() over(order by p.total_purchases ) as "lowest purchasing country",
       total_orders, row_number() over(order by total_orders ) as "lowest ordering country"
From (
    (SELECT "Country", sum(purchase_price) as total_purchases
    FROM public."Online_Retail"
    group by "Country"
    order by total_purchases
    limit 5) p inner join
    (SELECT "Country", count( distinct "InvoiceNo") as total_orders
    FROM public."Online_Retail"
    group by "Country"
    order by total_orders
    limit 5) s
    on p."Country" = s."Country");
```

Data output | Messages | Query History | Notifications

	number of countries bigint
1	38

Data output | Messages | Query History | Notifications

	Country text	total_purchases numeric	lowest purchasing country bigint	total_orders bigint	lowest ordering country bigint
1	Saudi Arabia	131.17	1	2	3
2	RSA	1002.31	2	1	1
3	Brazil	1143.60	3	1	2

Second Query

The Query

- after we get three countries that have been participated in both the last 5 countries in total purchasing and ordering and two of them are made only one order which is RSA and Brazil . the last one is saudi arabia which made 2 orders.
- now let's be sure its also inactive by finding the time difference between the first and the last date the saudi-arabia made their orders.
- First i created a VIEW called lowest_countries_vu contain the first query to have the three countries. then calculate the time differance.

```
select distinct "Country", first_value("InvoiceDate") over(partition by
"Country" order by "InvoiceDate" ) as forder_date,
first_value("InvoiceDate") over(partition by "Country" order by
"InvoiceDate" desc) as lorder_date,
first_value("InvoiceDate") over(partition by "Country" order by
"InvoiceDate" desc) -
first_value("InvoiceDate") over(partition by "Country" order by
"InvoiceDate" ) as "time between first and last order"
from public."Online_Retail"
where "Country" is not null
and "Country" in (
    select "Country" from lowest_countries_vu where "Country" = 'Saudi
Arabia' )
order by "time between first and last order";
```

Result

```
20 create or replace view lowest_countries_vu as (
21 select p."Country", total_Purchases, row_number() over(order by p.total_purchases ) as "lowest purchasing country",
22        total_orders, row_number() over(order by total_orders ) as "lowest ordering country"
23 From (
24     (SELECT "Country", sum(purchase_price) as total_purchases
25     FROM public."Online_Retail"
26     group by "Country"
27     order by total_purchases
28     limit 5) p inner join
29     (SELECT "Country", count( distinct "InvoiceNo") as total_orders
30     FROM public."Online_Retail"
31     group by "Country"
32     order by total_orders
33     limit 5) s
34 on p."Country" = s."Country"))
35
```

Data output | Messages | Query History | Notifications

```
CREATE VIEW

Query returned successfully in 137 msec.
```

Data output | Messages | Query History | Notifications

	Country text	forder_date date	lorder_date date	time between first and last order integer
1	Saudi Arabia	2011-02-24	2011-03-03	7

Third Query

Third Query

- now we want to classify them to levels depending on the total purchasing they did (the money we get). we don't have this information in our data set, so I created a new column called price_purching which equal (quantity * unit price).

```
alter table "Online_Retail"
add column purchase_price numeric ;
update "Online_Retail"
set purchase_price = "Quantity" * "UnitPrice";
```

- we will classify them to 5 categories

```
select *, NTILE(5) over(order by total_purchases desc) as
level
from (SELECT "Country", sum(purchase_price) as
total_purchases
    FROM public."Online_Retail"
        group by "Country" ) s;
```

Result

Data output | Messages | Query History | Notifications

	Country text	total_purchases numeric	level integer
1	United Kingdom	8187806.364	1
2	Netherlands	284661.54	1
3	EIRE	263276.82	1
4	Germany	221698.21	1
5	France	197403.90	1
6	Australia	137077.27	1
7	Switzerland	56385.35	1
8	Spain	54774.58	1
9	Belgium	40910.96	2
10	Sweden	36595.91	2
11	Japan	35340.62	2
12	Norway	35163.46	2
13	Portugal	29367.02	2
14	Finland	22326.74	2
15	Channel Islands	20086.29	2
16	Denmark	18768.14	2
17	Italy	16890.51	3

Total rows: 38 of 38 | Query complete 00:00:00.217

Fourth Query

The Query

- now we will focus only on the first level - we want to know the maimum, minimum deals or orders per day and the date for each value of them, for every country in the first level to be always ready for any load

```
select distinct "Country",
    first_value("orders per day") over (partition by "Country" order by "orders per day" desc) as "maximum orders per day",
    first_value("orders per day") over (partition by "Country" order by "orders per day" ) as "minimum orders per day",
    first_value("InvoiceDate") over (partition by "Country" order by "orders per day" desc) as "date of maximum orders per
day",
    first_value("InvoiceDate") over (partition by "Country" order by "orders per day" ) as "date of minimum orders per day"
from
   (
    select distinct ("Country"), "InvoiceDate", count("InvoiceNo") over (partition by "Country" order by "InvoiceDate"
           range between interval '12' hour preceding and interval '12' hour following) as "orders per day"
    from public."Online_Retail"
    where "Country" in (
       select x."Country"
       from (
            select s.*, NTILE(5) over(order by total_purchases desc) AS level
            from (
                SELECT "Country", sum(purchase_price) as total_purchases
                FROM public."Online_Retail"
                group by "Country"
                ) s
           ) x
                where x.level = 1)) v
order by "maximum orders per day" desc ;
```

Result

Data output | Messages | Query History | Notifications

	Country text	max ord pday bigint	min ord pday bigint	date max ord pday date	date of min ord pday date
1	United Kingdom	5200	250	2011-12-05	2010-12-22
2	France	262	1	2011-10-11	2011-02-14
3	Germany	228	1	2011-01-07	2011-04-06
4	Switzerland	223	1	2011-08-02	2011-07-15
5	EIRE	188	1	2011-09-07	2011-07-08
6	Netherlands	146	1	2011-10-20	2011-02-03
7	Australia	139	1	2011-06-15	2011-04-01
8	Spain	122	1	2011-08-28	2011-10-04

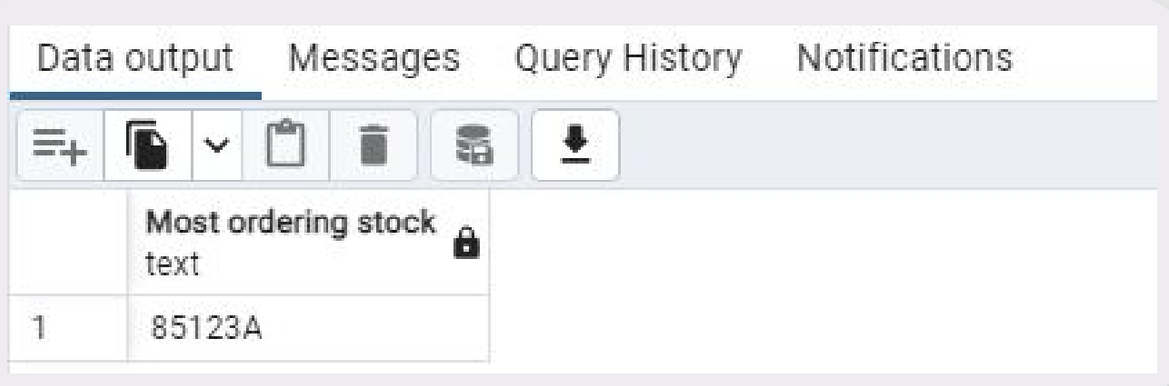
Fifth Query

The Query

Result

- after we knew the max and min number of orders, it's time to know the most stock that get orders from those first level countries to prepare it.

```
select  distinct first_value("StockCode") over(order by total_orders DESC ) as
"Most ordering stock"
from (SELECT "Country","StockCode", count( distinct "InvoiceNo") as total_orders
FROM public."Online_Retail"
group by "Country","StockCode" ) s
where "Country" in (
        select x."Country"
        from (
             select s.*, NTILE(5) over(order by total_purchases desc) AS level
             from (
                  SELECT "Country", sum(purchase_price) as total_purchases
                  FROM public."Online_Retail"
                  group by "Country"
                  ) s
            ) x
                  where x.level = 1);
```



Sixth Query

The Query

- **now we want to know the most 10 purchasing customers in the first level countries to give them a prize, and how percentage that every one of them have a customers had a purchases more than him over the whole customers.**

```
select *, rank() over(order by total_purchases desc), percent_rank() over(order by
total_purchases desc )
from (SELECT  "Country", "CustomerID", sum(purchase_price) as total_purchases
FROM public."Online_Retail"
group by "Country","CustomerID" ) s
where "Country" in (
        select x."Country"
        from (
             select s.*, NTILE(5) over(order by total_purchases desc) AS level
             from (
                  SELECT "Country", sum(purchase_price) as total_purchases
                  FROM public."Online_Retail"
                  group by "Country"
                  ) s
            ) x
                  where x.level = 1)
limit 10;
```

Result

Data output | Messages | Query History | Notifications

	Country text	CustomerID character varying	total_purchases numeric	rank bigint	percent_rank double precision
1	United Kingdom		1419932.97	1	0
2	Netherlands	14646	279489.02	2	0.0002376425855513308
3	United Kingdom	18102	256438.49	3	0.0004752851711026616
4	United Kingdom	17450	187482.17	4	0.0007129277566539924
5	EIRE	14911	132572.62	5	0.0009505703422053232
6	Australia	12415	123725.45	6	0.001188212927756654
7	EIRE	14156	113384.14	7	0.0014258555133079848
8	United Kingdom	17511	88125.38	8	0.0016634980988593155
9	United Kingdom	16684	65892.08	9	0.0019011406844106464
10	United Kingdom	13694	62653.10	10	0.0021387832699619773