Data Findings:

Table	#Customers	#Orders	Sales
Sales	**33,751**	**33,751**	**49,159,193.52**
Returns	**15,828**	**15,828**	**12,772,338.60**

1) SALES TABLE HAS NO DUPLICATES PRESENT WHILE RETURNS TABLE HAS **54** DUPLICATE RECORDS.

2) SALES TABLE HAS **29,141** UNIQUE CUSTOMER ID'S WHILE RETURNS TABLE HAS **13,487** UNIQUE CUSTOMER ID'S.

3) SALES TABLE HAS **33,751** UNIQUE ORDER ID'S WHILE RETURNS TABLE HAS **14,417** UNIQUE ORDER ID'S.

4) TRANSACTION DATE RANGES FROM **JAN-1-2015 TO JAN-15-2015.** RETURN DATE RANGES FROM **DEC-30-2014 TO SEPT-9-2016**.

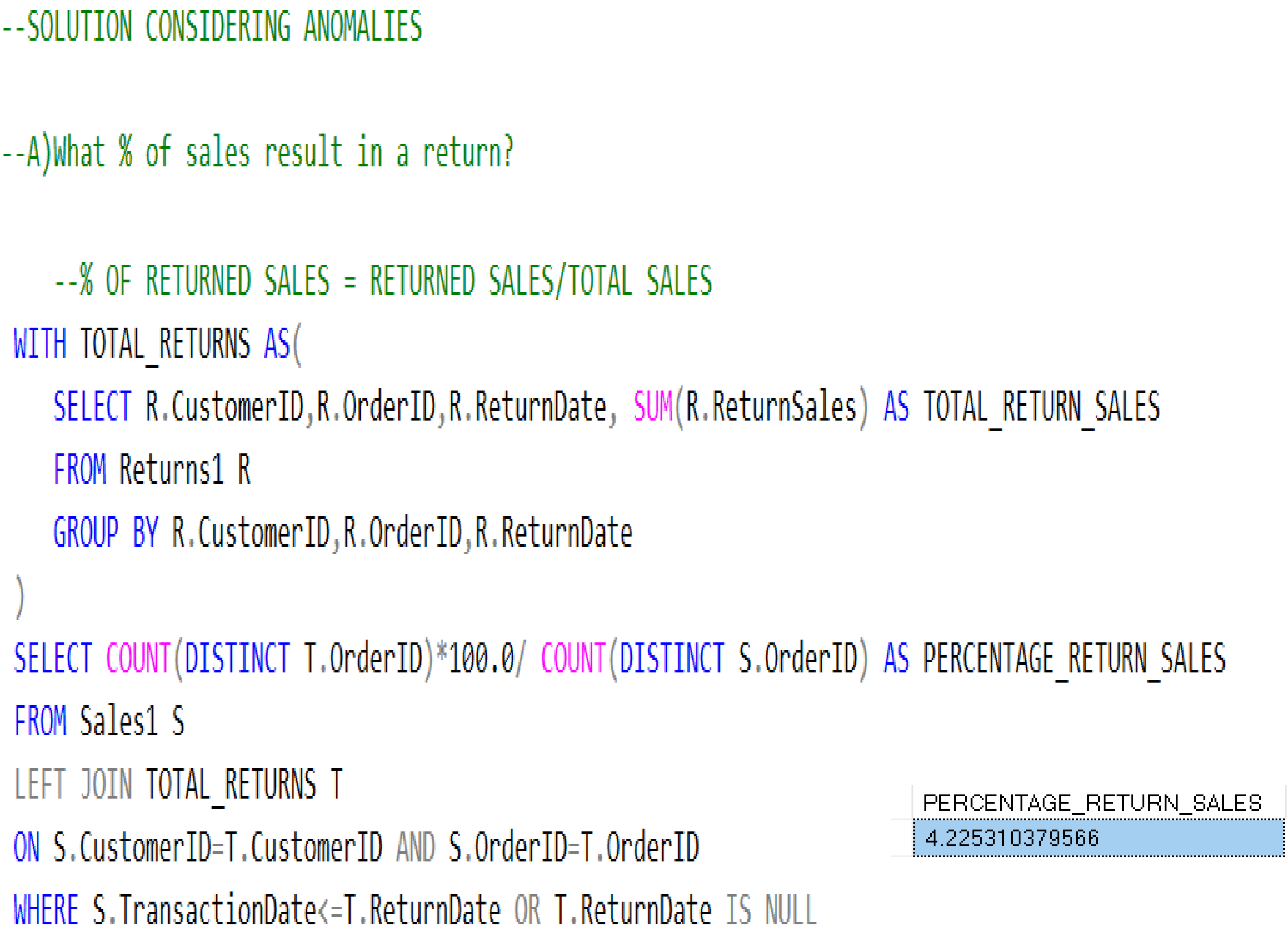
5) **3,798** CUSTOMERS PLACED MORE THAN 1 ORDER. **1,851** CUSTOMERS RETURNED MORE THAN 1 ORDER.

6) RETURN DATE HAS NO TIME CONSTRAINT. **602** ORDERS ARE RETURNED MORE THAN 30 DAYS AFTER PURCHASING.

7) **2** ORDERS HAVE RETURN SALES > SALES.

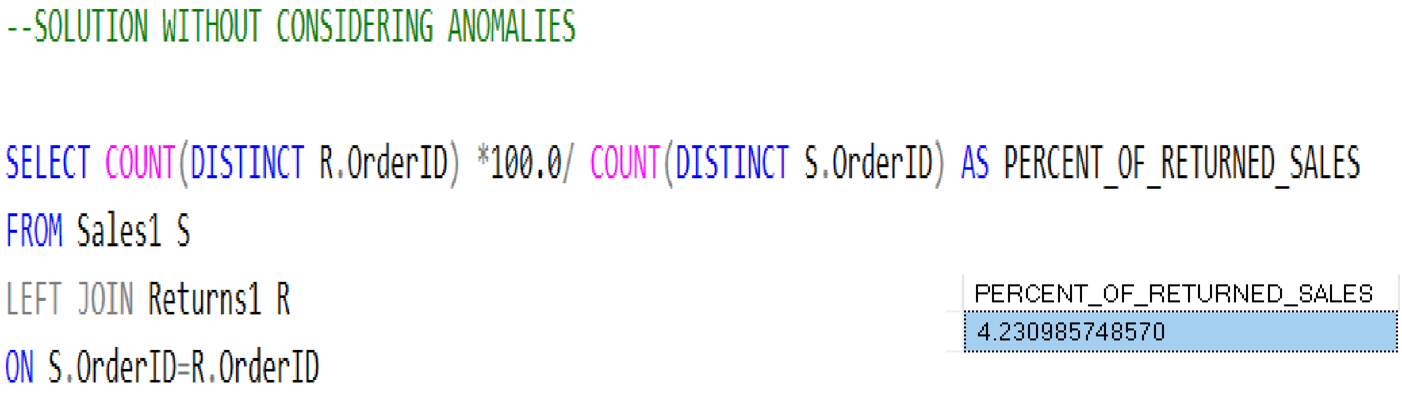
Data Anomalies:

	#RECORDS
1) ORDER ID IS NOT UNIQUE IN RETURNS TABLE, FEW CUSTOMERS ARE ASSIGNED SAME ORDER ID	**47**
2) CUSTOMERS RETURNED SAME ORDER ID MORE THAN ONCE.	**1203**
3) RECORDS WITH RETURNED DATE < TRANSACTION DATE.	**3**
4) RECORDS WITH DIFFERENT CUSTOMER ID FOR SAME ORDERS IN SALES AND RETURNS TABLE RESPECTIVELY.	**4**
5) RETURN DATE HAS NULL VALUES.	**20**
6) RECORDS WITH MULTIPLE RETURN SALES FOR SAME CUSTOMER IDs AND ORDER IDs. ASSUMING THESE AS PARTIAL RETURNS.	**1310**



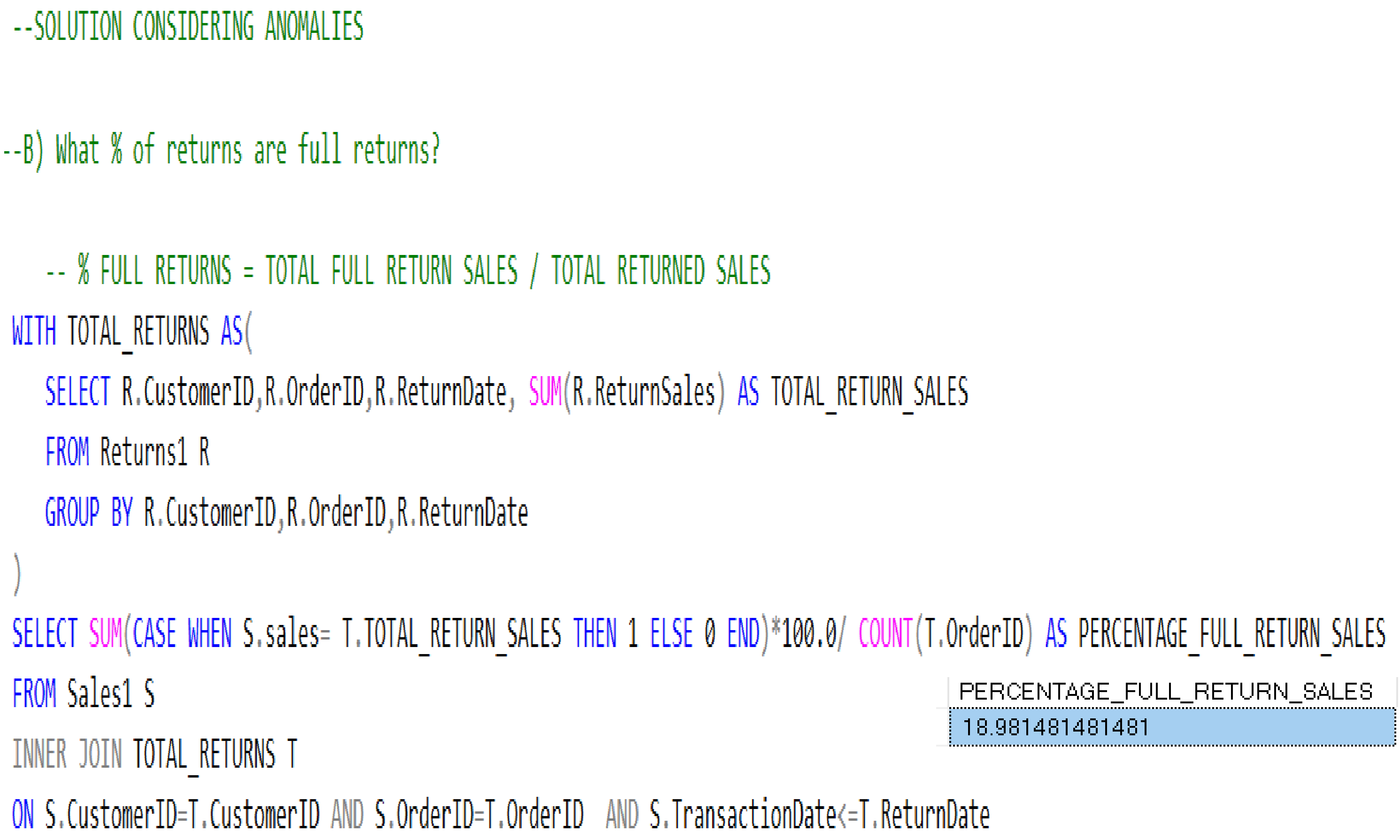
```
--SOLUTION CONSIDERING ANOMALIES

--A)What % of sales result in a return?

    --% OF RETURNED SALES = RETURNED SALES/TOTAL SALES
 WITH TOTAL_RETURNS AS(
    SELECT R.CustomerID,R.OrderID,R.ReturnDate, SUM(R.ReturnSales) AS TOTAL_RETURN_SALES
    FROM Returns1 R
    GROUP BY R.CustomerID,R.OrderID,R.ReturnDate
 )
 SELECT COUNT(DISTINCT T.OrderID)*100.0/ COUNT(DISTINCT S.OrderID) AS PERCENTAGE_RETURN_SALES
 FROM Sales1 S
 LEFT JOIN TOTAL_RETURNS T
 ON S.CustomerID=T.CustomerID AND S.OrderID=T.OrderID
 WHERE S.TransactionDate<=T.ReturnDate OR T.ReturnDate IS NULL
```

PERCENTAGE_RETURN_SALES
4.225310379566



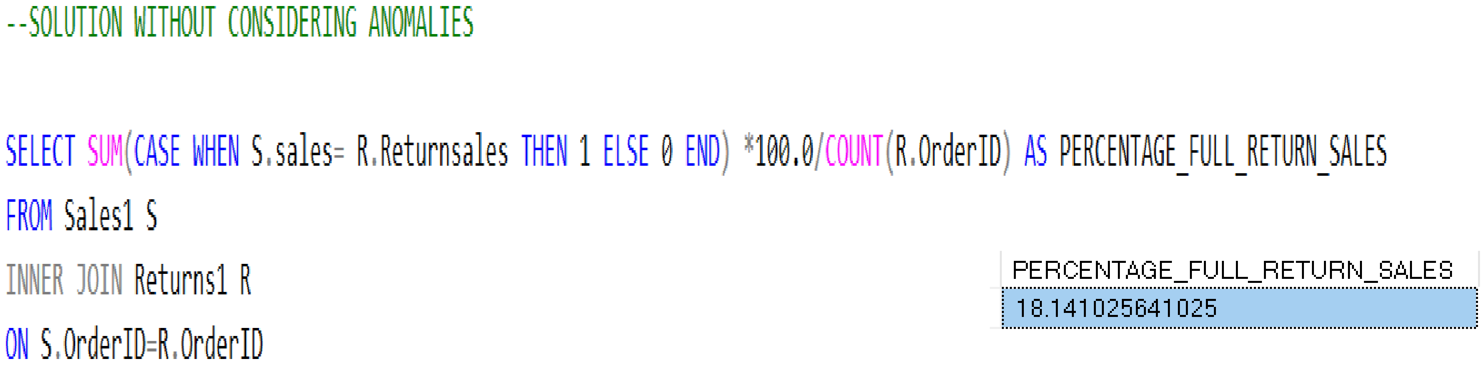
```
 --SOLUTION WITHOUT CONSIDERING ANOMALIES

 SELECT COUNT(DISTINCT R.OrderID) *100.0/ COUNT(DISTINCT S.OrderID) AS PERCENT_OF_RETURNED_SALES
 FROM Sales1 S
 LEFT JOIN Returns1 R
 ON S.OrderID=R.OrderID
```

PERCENT_OF_RETURNED_SALES
4.230985748570



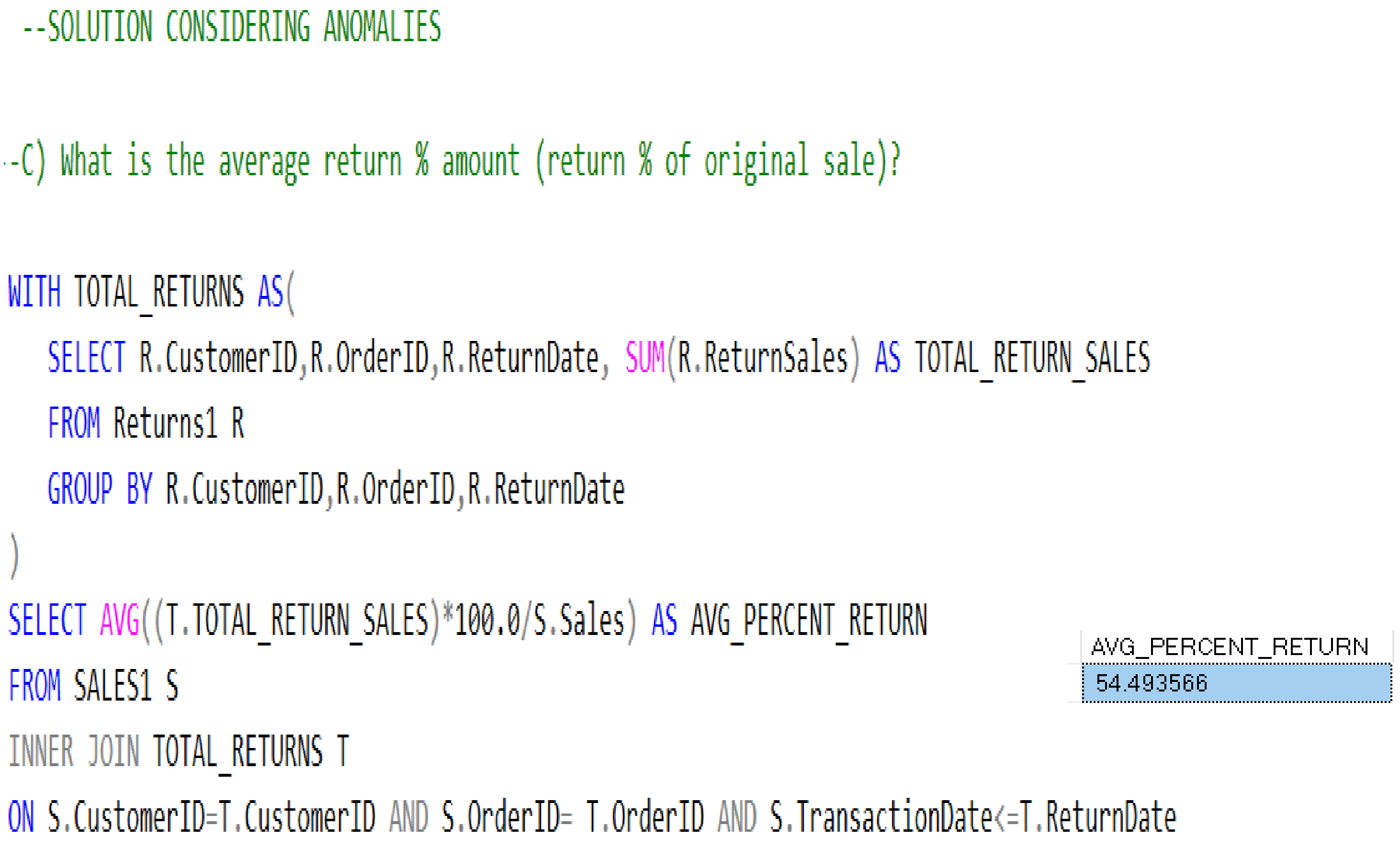
```
--SOLUTION CONSIDERING ANOMALIES

--B) What % of returns are full returns?

   -- % FULL RETURNS = TOTAL FULL RETURN SALES / TOTAL RETURNED SALES
WITH TOTAL_RETURNS AS(
   SELECT R.CustomerID,R.OrderID,R.ReturnDate, SUM(R.ReturnSales) AS TOTAL_RETURN_SALES
   FROM Returns1 R
   GROUP BY R.CustomerID,R.OrderID,R.ReturnDate
)
SELECT SUM(CASE WHEN S.sales= T.TOTAL_RETURN_SALES THEN 1 ELSE 0 END)*100.0/ COUNT(T.OrderID) AS PERCENTAGE_FULL_RETURN_SALES
FROM Sales1 S
INNER JOIN TOTAL_RETURNS T
ON S.CustomerID=T.CustomerID AND S.OrderID=T.OrderID  AND S.TransactionDate<=T.ReturnDate
```

PERCENTAGE_FULL_RETURN_SALES
18.981481481481



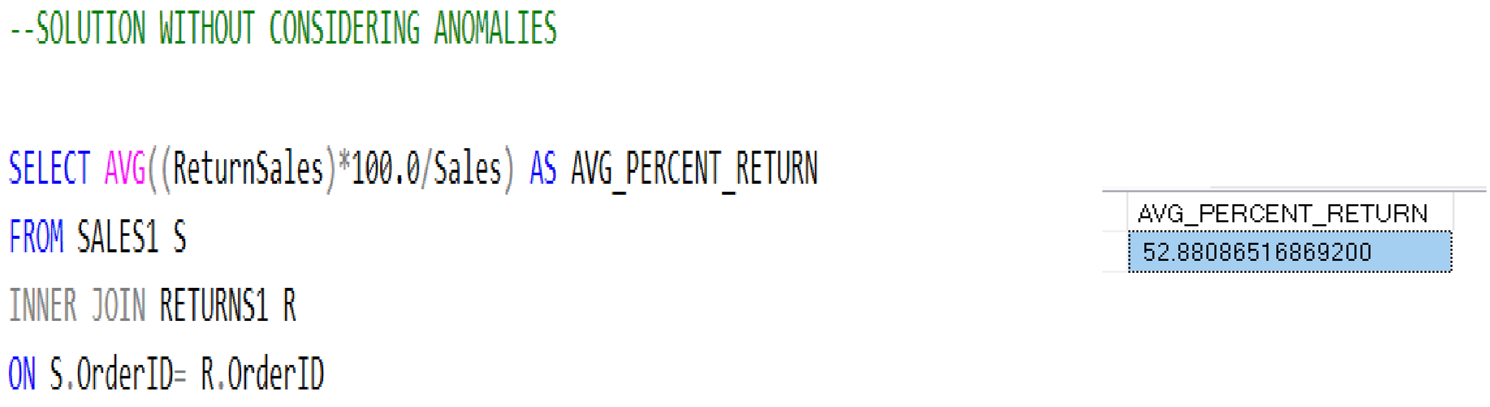
```
--SOLUTION WITHOUT CONSIDERING ANOMALIES

SELECT SUM(CASE WHEN S.sales= R.Returnsales THEN 1 ELSE 0 END) *100.0/COUNT(R.OrderID) AS PERCENTAGE_FULL_RETURN_SALES
FROM Sales1 S
INNER JOIN Returns1 R
ON S.OrderID=R.OrderID
```

PERCENTAGE_FULL_RETURN_SALES
18.141025641025



```
 --SOLUTION CONSIDERING ANOMALIES

-C) What is the average return % amount (return % of original sale)?

WITH TOTAL_RETURNS AS(
   SELECT R.CustomerID,R.OrderID,R.ReturnDate, SUM(R.ReturnSales) AS TOTAL_RETURN_SALES
   FROM Returns1 R
   GROUP BY R.CustomerID,R.OrderID,R.ReturnDate
)
SELECT AVG((T.TOTAL_RETURN_SALES)*100.0/S.Sales) AS AVG_PERCENT_RETURN
FROM SALES1 S
INNER JOIN TOTAL_RETURNS T
ON S.CustomerID=T.CustomerID AND S.OrderID= T.OrderID AND S.TransactionDate<=T.ReturnDate
```

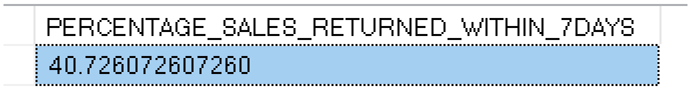
AVG_PERCENT_RETURN
54.493566



```
--SOLUTION WITHOUT CONSIDERING ANOMALIES

SELECT AVG((ReturnSales)*100.0/Sales) AS AVG_PERCENT_RETURN
FROM SALES1 S
INNER JOIN RETURNS1 R
ON S.OrderID= R.OrderID
```

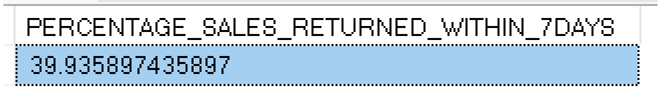
AVG_PERCENT_RETURN
52.88086516869200

```sql
--SOLUTION CONSIDERING ANOMALIES

--D) What % of returns occur within 7 days of the original sale?

   -- = NO.OF RETURNS WITHIN 7 DAYS/ TOTAL RETURNED ORDERS
WITH TOTAL_RETURNS AS(
   SELECT R.CustomerID,R.OrderID,R.ReturnDate, SUM(R.ReturnSales) AS TOTAL_RETURN_SALES
   FROM Returns1 R
   GROUP BY R.CustomerID,R.OrderID,R.ReturnDate
)
SELECT SUM(CASE
   WHEN T.ReturnDate BETWEEN S.TransactionDate AND DATEADD(DAY, 7,S.TransactionDate) THEN  1 ELSE 0
   END)*100.0/ COUNT(T.OrderID) AS PERCENTAGE_SALES_RETURNED_WITHIN_7DAYS
FROM Sales1 S
INNER JOIN TOTAL_RETURNS T
ON S.CustomerID=T.CustomerID AND S.OrderID=T.OrderID
```

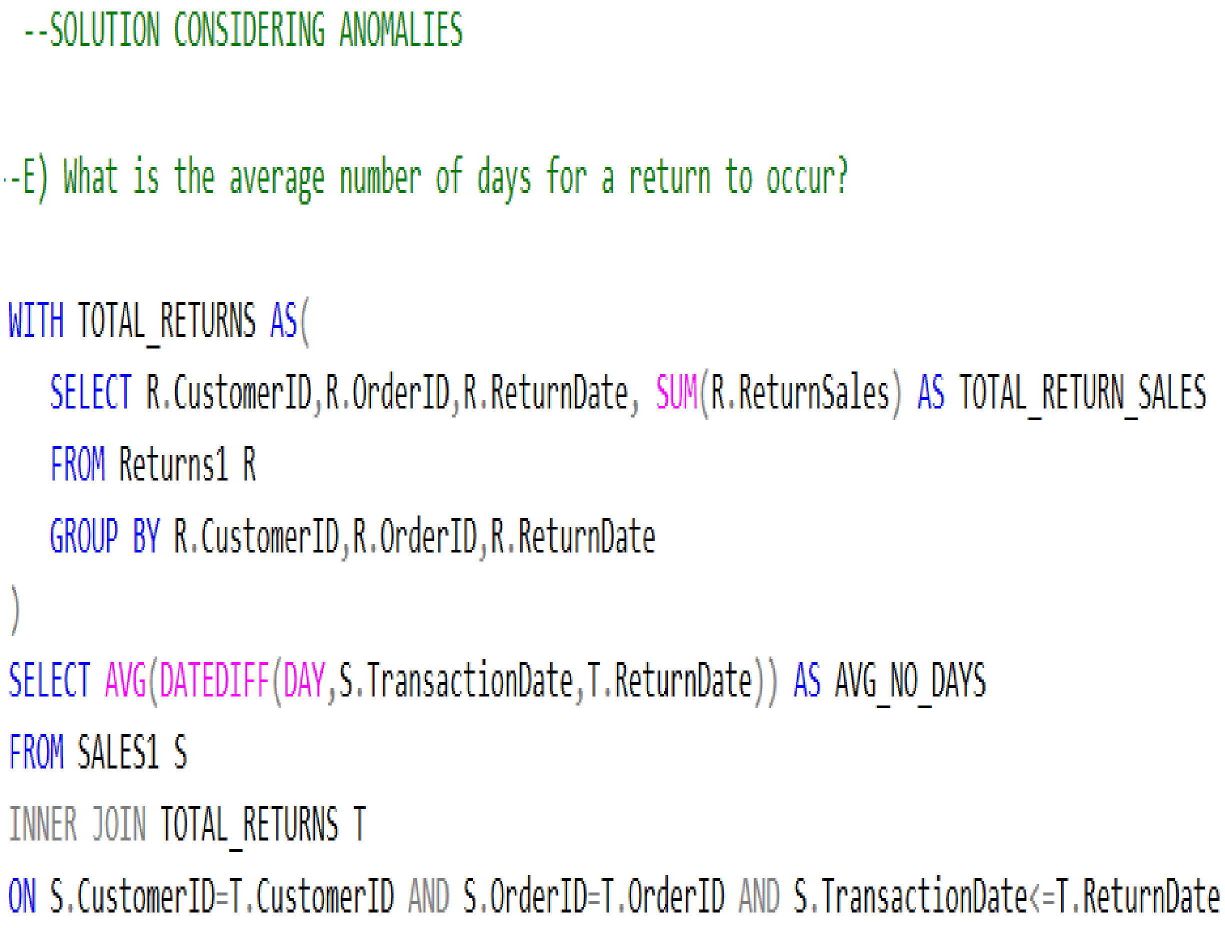


```
PERCENTAGE_SALES_RETURNED_WITHIN_7DAYS
40.726072607260
```

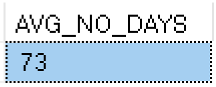
```sql
--SOLUTION WITHOUT CONSIDERING ANOMALIES

SELECT SUM(CASE
WHEN R.ReturnDate BETWEEN S.TransactionDate AND DATEADD(DAY, 7,S.TransactionDate) THEN 1 ELSE 0
END)*100.0/ COUNT(R.OrderID) AS PERCENTAGE_SALES_RETURNED_WITHIN_7DAYS
FROM Sales1 S
INNER JOIN Returns1 R
ON S.OrderID=R.OrderID
```



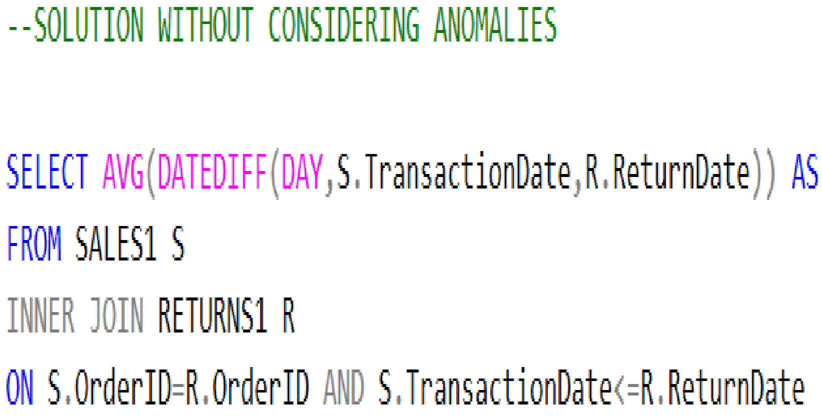
```
PERCENTAGE_SALES_RETURNED_WITHIN_7DAYS
39.935897435897
```



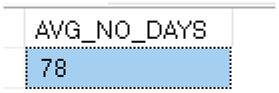
```
 --SOLUTION CONSIDERING ANOMALIES

--E) What is the average number of days for a return to occur?

WITH TOTAL_RETURNS AS(
   SELECT R.CustomerID,R.OrderID,R.ReturnDate, SUM(R.ReturnSales) AS TOTAL_RETURN_SALES
   FROM Returns1 R
   GROUP BY R.CustomerID,R.OrderID,R.ReturnDate
)
SELECT AVG(DATEDIFF(DAY,S.TransactionDate,T.ReturnDate)) AS AVG_NO_DAYS
FROM SALES1 S
INNER JOIN TOTAL_RETURNS T
ON S.CustomerID=T.CustomerID AND S.OrderID=T.OrderID AND S.TransactionDate<=T.ReturnDate
```



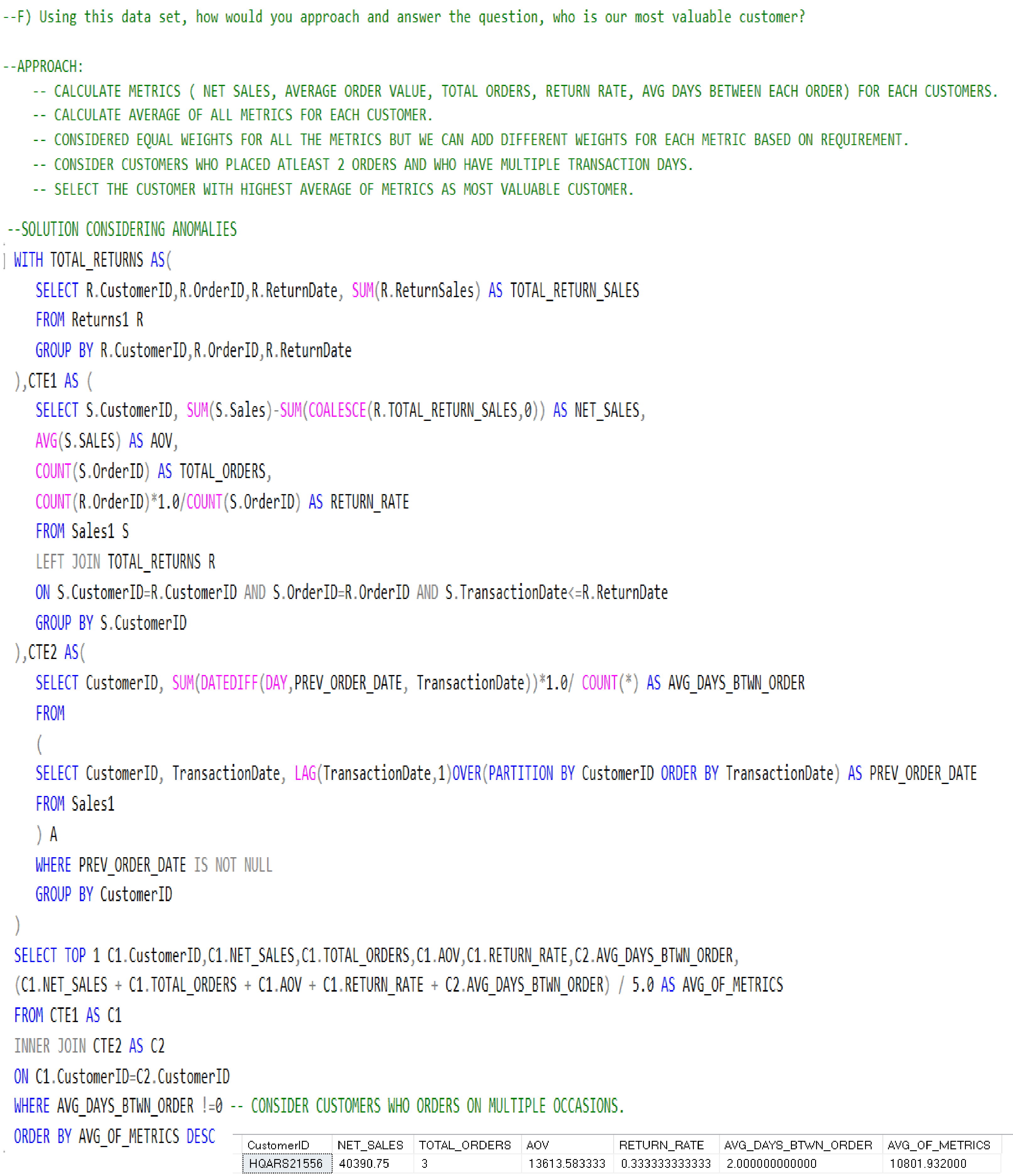
AVG_NO_DAYS
73



```
--SOLUTION WITHOUT CONSIDERING ANOMALIES

SELECT AVG(DATEDIFF(DAY,S.TransactionDate,R.ReturnDate)) AS AVG_NO_DAYS
FROM SALES1 S
INNER JOIN RETURNS1 R
ON S.OrderID=R.OrderID AND S.TransactionDate<=R.ReturnDate
```

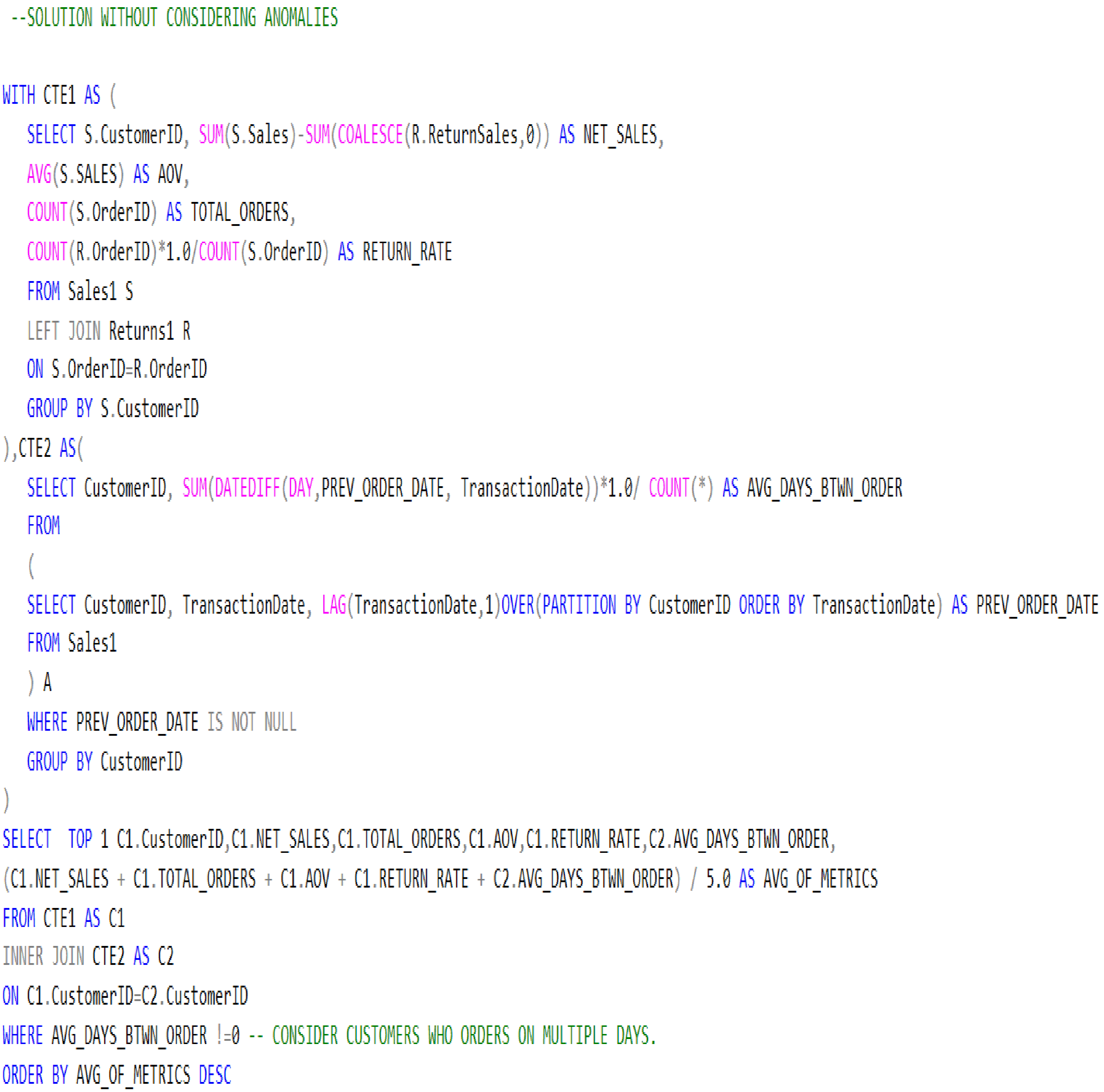


AVG_NO_DAYS
78

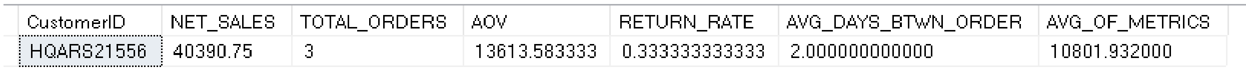


```
--F) Using this data set, how would you approach and answer the question, who is our most valuable customer?

--APPROACH:
    -- CALCULATE METRICS ( NET SALES, AVERAGE ORDER VALUE, TOTAL ORDERS, RETURN RATE, AVG DAYS BETWEEN EACH ORDER) FOR EACH CUSTOMERS.
    -- CALCULATE AVERAGE OF ALL METRICS FOR EACH CUSTOMER.
    -- CONSIDERED EQUAL WEIGHTS FOR ALL THE METRICS BUT WE CAN ADD DIFFERENT WEIGHTS FOR EACH METRIC BASED ON REQUIREMENT.
    -- CONSIDER CUSTOMERS WHO PLACED ATLEAST 2 ORDERS AND WHO HAVE MULTIPLE TRANSACTION DAYS.
    -- SELECT THE CUSTOMER WITH HIGHEST AVERAGE OF METRICS AS MOST VALUABLE CUSTOMER.

 --SOLUTION CONSIDERING ANOMALIES
 WITH TOTAL_RETURNS AS(
    SELECT R.CustomerID,R.OrderID,R.ReturnDate, SUM(R.ReturnSales) AS TOTAL_RETURN_SALES
    FROM Returns1 R
    GROUP BY R.CustomerID,R.OrderID,R.ReturnDate
 ),CTE1 AS (
    SELECT S.CustomerID, SUM(S.Sales)-SUM(COALESCE(R.TOTAL_RETURN_SALES,0)) AS NET_SALES,
    AVG(S.SALES) AS AOV,
    COUNT(S.OrderID) AS TOTAL_ORDERS,
    COUNT(R.OrderID)*1.0/COUNT(S.OrderID) AS RETURN_RATE
    FROM Sales1 S
    LEFT JOIN TOTAL_RETURNS R
    ON S.CustomerID=R.CustomerID AND S.OrderID=R.OrderID AND S.TransactionDate<=R.ReturnDate
    GROUP BY S.CustomerID
 ),CTE2 AS(
    SELECT CustomerID, SUM(DATEDIFF(DAY,PREV_ORDER_DATE, TransactionDate))*1.0/ COUNT(*) AS AVG_DAYS_BTWN_ORDER
    FROM
    (
    SELECT CustomerID, TransactionDate, LAG(TransactionDate,1)OVER(PARTITION BY CustomerID ORDER BY TransactionDate) AS PREV_ORDER_DATE
    FROM Sales1
    ) A
    WHERE PREV_ORDER_DATE IS NOT NULL
    GROUP BY CustomerID
 )
 SELECT TOP 1 C1.CustomerID,C1.NET_SALES,C1.TOTAL_ORDERS,C1.AOV,C1.RETURN_RATE,C2.AVG_DAYS_BTWN_ORDER,
 (C1.NET_SALES + C1.TOTAL_ORDERS + C1.AOV + C1.RETURN_RATE + C2.AVG_DAYS_BTWN_ORDER) / 5.0 AS AVG_OF_METRICS
 FROM CTE1 AS C1
 INNER JOIN CTE2 AS C2
 ON C1.CustomerID=C2.CustomerID
 WHERE AVG_DAYS_BTWN_ORDER !=0 -- CONSIDER CUSTOMERS WHO ORDERS ON MULTIPLE OCCASIONS.
 ORDER BY AVG_OF_METRICS DESC
```

CustomerID	NET_SALES	TOTAL_ORDERS	AOV	RETURN_RATE	AVG_DAYS_BTWN_ORDER	AVG_OF_METRICS
HQARS21556	40390.75	3	13613.583333	0.333333333333	2.000000000000	10801.932000



```
 --SOLUTION WITHOUT CONSIDERING ANOMALIES

WITH CTE1 AS (
   SELECT S.CustomerID, SUM(S.Sales)-SUM(COALESCE(R.ReturnSales,0)) AS NET_SALES,
   AVG(S.SALES) AS AOV,
   COUNT(S.OrderID) AS TOTAL_ORDERS,
   COUNT(R.OrderID)*1.0/COUNT(S.OrderID) AS RETURN_RATE
   FROM Sales1 S
   LEFT JOIN Returns1 R
   ON S.OrderID=R.OrderID
   GROUP BY S.CustomerID
),CTE2 AS(
   SELECT CustomerID, SUM(DATEDIFF(DAY,PREV_ORDER_DATE, TransactionDate))*1.0/ COUNT(*) AS AVG_DAYS_BTWN_ORDER
   FROM
   (
   SELECT CustomerID, TransactionDate, LAG(TransactionDate,1)OVER(PARTITION BY CustomerID ORDER BY TransactionDate) AS PREV_ORDER_DATE
   FROM Sales1
   ) A
   WHERE PREV_ORDER_DATE IS NOT NULL
   GROUP BY CustomerID
)
SELECT  TOP 1 C1.CustomerID,C1.NET_SALES,C1.TOTAL_ORDERS,C1.AOV,C1.RETURN_RATE,C2.AVG_DAYS_BTWN_ORDER,
(C1.NET_SALES + C1.TOTAL_ORDERS + C1.AOV + C1.RETURN_RATE + C2.AVG_DAYS_BTWN_ORDER) / 5.0 AS AVG_OF_METRICS
FROM CTE1 AS C1
INNER JOIN CTE2 AS C2
ON C1.CustomerID=C2.CustomerID
WHERE AVG_DAYS_BTWN_ORDER !=0 -- CONSIDER CUSTOMERS WHO ORDERS ON MULTIPLE DAYS.
ORDER BY AVG_OF_METRICS DESC
```



CustomerID	NET_SALES	TOTAL_ORDERS	AOV	RETURN_RATE	AVG_DAYS_BTWN_ORDER	AVG_OF_METRICS
HQARS21556	40390.75	3	13613.583333	0.333333333333	2.000000000000	10801.932000