/\* Analyze the data by finding the answers to the questions below:

1.Find the top 3 customers who have the maximum count of orders.

\*/

SELECT

TOP 3 \*

FROM(

SELECT

DISTINCT B.cust\_id, first\_name, last\_name,

COUNT(ord\_id) OVER(PARTITION BY B.cust\_id) AS cnt\_order

FROM

customer.customer\_table AS A,

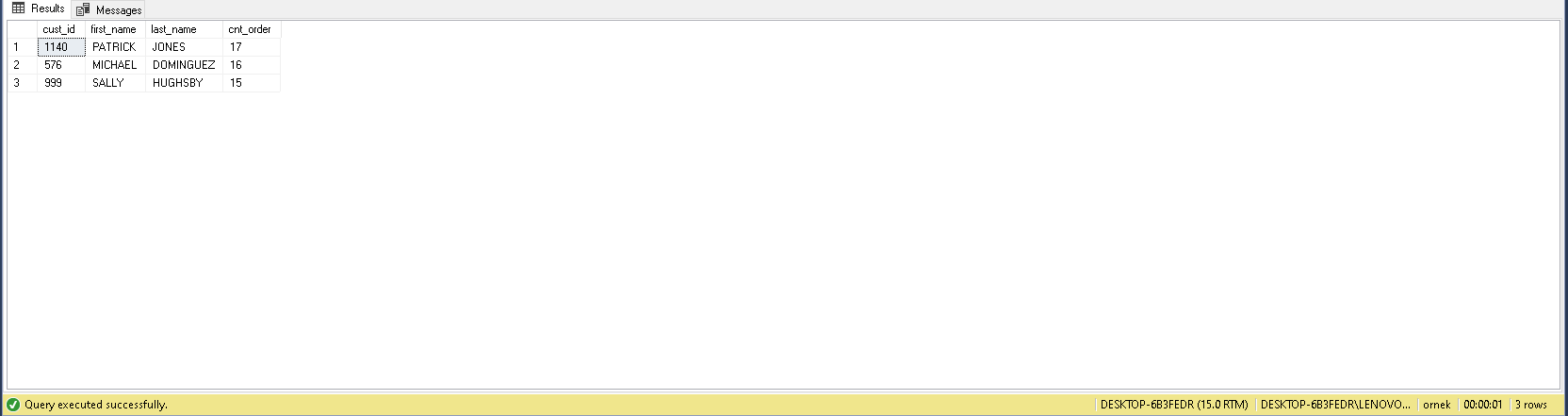
[order].order\_table AS B

WHERE

A.cust\_id = B.cust\_id

) AS Subquery

ORDER BY cnt\_order DESC



/\* 2. Find the customer whose order took the maximum time to get shipping. \*/

WITH T1 AS(

SELECT

A.first\_name,

A.last\_name,

C.days\_taken\_for\_shipping

FROM

customer.customer\_table AS A

INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id

INNER JOIN customer.ship\_table AS C ON C.ship\_id = B.ship\_id)

SELECT

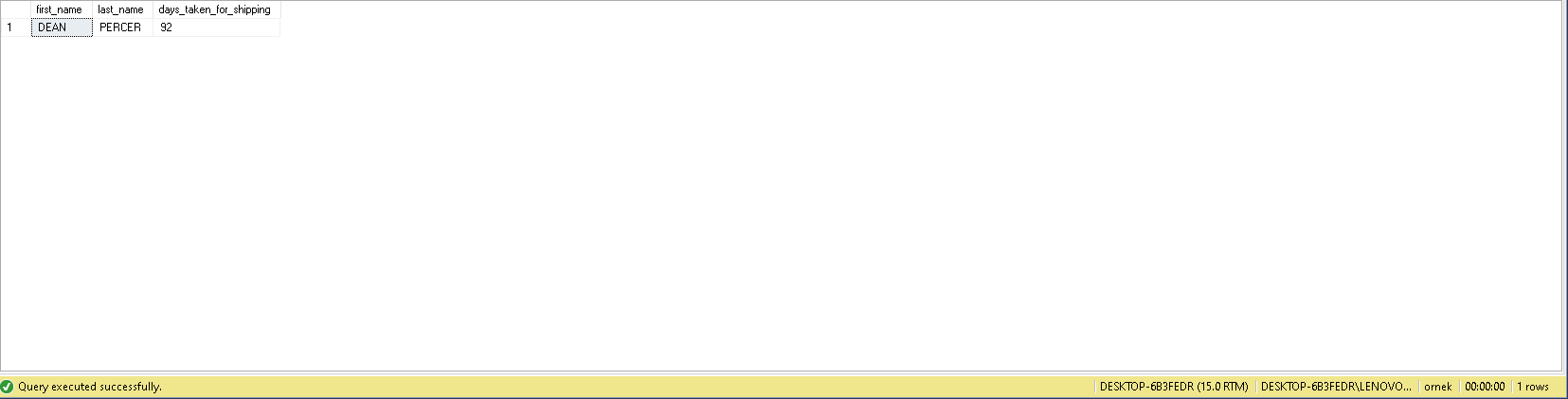
TOP 1 \*

FROM

T1

ORDER BY

days\_taken\_for\_shipping DESC



/\* 3. Count the total number of unique customers in January and how many of them came back every month over the entire year in 2011 \*/

WITH T1 AS(

SELECT

A.\*,

DATENAME(MONTH, B.order\_date) AS month\_name

FROM

customer.customer\_table AS A

INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id

WHERE

YEAR(B.order\_date) = 2011 AND

A.cust\_id IN(

SELECT

DISTINCT A.cust\_id

FROM

customer.customer\_table AS A

INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id

WHERE

YEAR(B.order\_date) = 2011 AND MONTH(B.order\_date) = 1 ))

SELECT

SUM(CASE WHEN month\_name = 'January' THEN 1 ELSE 0 END) AS January,

SUM(CASE WHEN month\_name = 'February' THEN 1 ELSE 0 END) AS February,

SUM(CASE WHEN month\_name = 'March' THEN 1 ELSE 0 END) AS March,

SUM(CASE WHEN month\_name = 'April' THEN 1 ELSE 0 END) AS April,

SUM(CASE WHEN month\_name = 'May' THEN 1 ELSE 0 END) AS May,

SUM(CASE WHEN month\_name = 'June' THEN 1 ELSE 0 END) AS June,

SUM(CASE WHEN month\_name = 'July' THEN 1 ELSE 0 END) AS July,

SUM(CASE WHEN month\_name = 'August' THEN 1 ELSE 0 END) AS August,

SUM(CASE WHEN month\_name = 'September' THEN 1 ELSE 0 END) AS September,

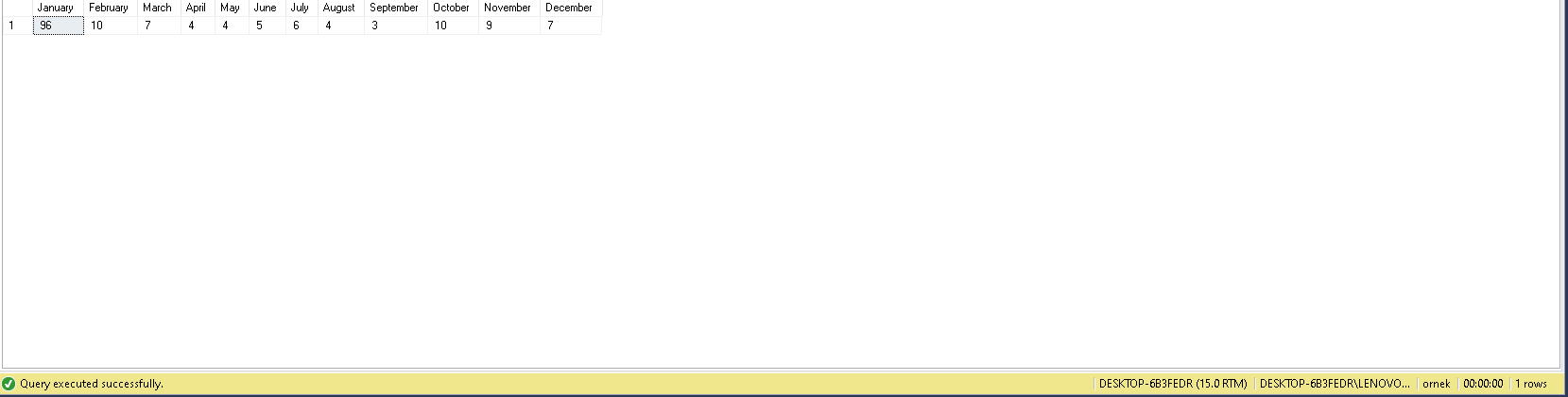
SUM(CASE WHEN month\_name = 'October' THEN 1 ELSE 0 END) AS October,

SUM(CASE WHEN month\_name = 'November' THEN 1 ELSE 0 END) AS November,

SUM(CASE WHEN month\_name = 'December' THEN 1 ELSE 0 END) AS December

FROM

T1



/\* 4. Write a query to return for each user the time elapsed between the first purchasing and the third purchasing, in ascending order by Customer ID. \*/

SELECT

\*,

DATEDIFF(DAY, first\_purchasing, third\_purchasing) AS difference\_first\_and\_third\_purchase

FROM(

SELECT

A.cust\_id,

order\_date AS first\_purchasing,

LEAD(order\_date, 2) OVER(PARTITION BY A.cust\_id ORDER BY B.order\_date) AS third\_purchasing

FROM

customer.customer\_table AS A

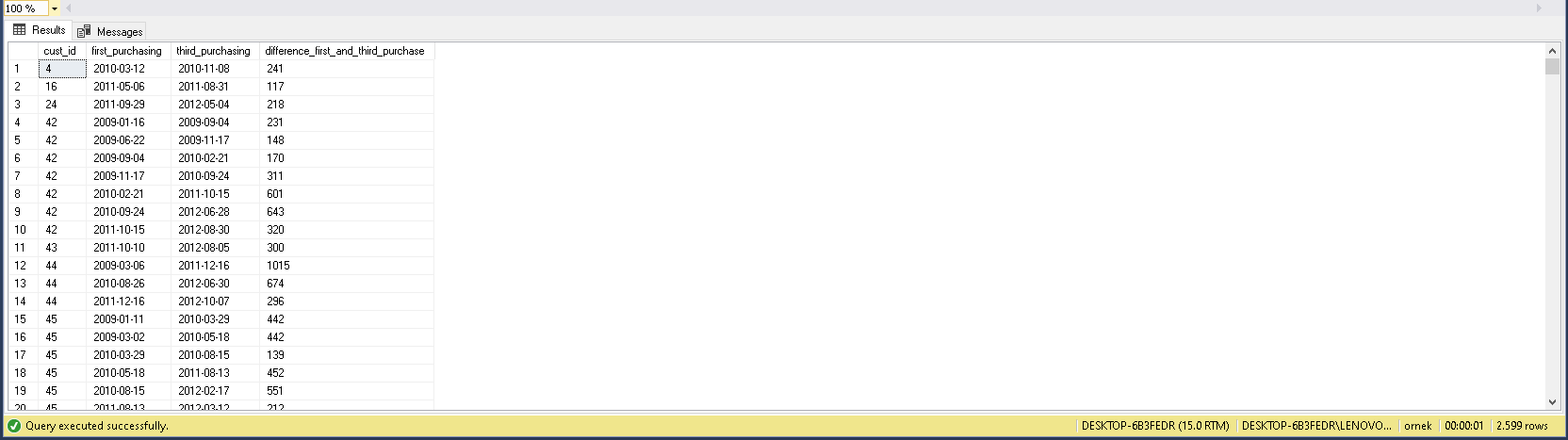
INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id) AS subquery

WHERE

third\_purchasing IS NOT NULL

ORDER BY

cust\_id;



/\* 5 . Write a query that returns customers who purchased both product 11 and product 14, as well as the ratio of these products to the total number of products purchased by the customer. \*/

GO

WITH T1 AS (

SELECT

DISTINCT A.cust\_id,

SUM(C.order\_quantity) OVER(PARTITION BY A.cust\_id) AS prod\_id\_11,

A.first\_name,

A.last\_name

---CASE WHEN C.prod\_id = 11 AND C.order\_quantity ! = 0 THEN SUM(order\_quantity) OVER(PARTITION BY A.cust\_id) ELSE 0 END AS prod\_id\_11

FROM

customer.customer\_table AS A

INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id

INNER JOIN [order].order\_item AS C ON C.ord\_id = B.ord\_id

WHERE

C.prod\_id = 11

), T2 AS(

SELECT

DISTINCT A.cust\_id,

SUM(C.order\_quantity) OVER(PARTITION BY A.cust\_id) AS prod\_id\_14,

A.first\_name,

A.last\_name

---CASE WHEN C.prod\_id = 14 AND C.order\_quantity ! = 0 THEN SUM(order\_quantity) OVER(PARTITION BY A.cust\_id) ELSE 0 END AS prod\_id\_14

FROM

customer.customer\_table AS A

INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id

INNER JOIN [order].order\_item AS C ON C.ord\_id = B.ord\_id

WHERE

C.prod\_id = 14

), T3 AS(

SELECT

DISTINCT A.cust\_id,

SUM(C.order\_quantity) OVER(PARTITION BY A.cust\_id) AS sum\_quantity,

A.first\_name,

A.last\_name

---CASE WHEN C.prod\_id = 14 AND C.order\_quantity ! = 0 THEN SUM(order\_quantity) OVER(PARTITION BY A.cust\_id) ELSE 0 END AS prod\_id\_14

FROM

customer.customer\_table AS A

INNER JOIN [order].order\_table AS B ON B.cust\_id = A.cust\_id

INNER JOIN [order].order\_item AS C ON C.ord\_id = B.ord\_id

)

SELECT

T1.cust\_id,

T1.first\_name,

T1.last\_name,

T3.sum\_quantity,

T1.prod\_id\_11,

CAST(T1.prod\_id\_11 \* 1.0 / T3.sum\_quantity \* 1.0 AS DECIMAL(3,2)) AS ratio\_of\_product\_11,

T2.prod\_id\_14,

CAST(T2.prod\_id\_14 \* 1.0 / T3.sum\_quantity \* 1.0 AS DECIMAL(3,2)) AS ratio\_of\_product\_14

FROM

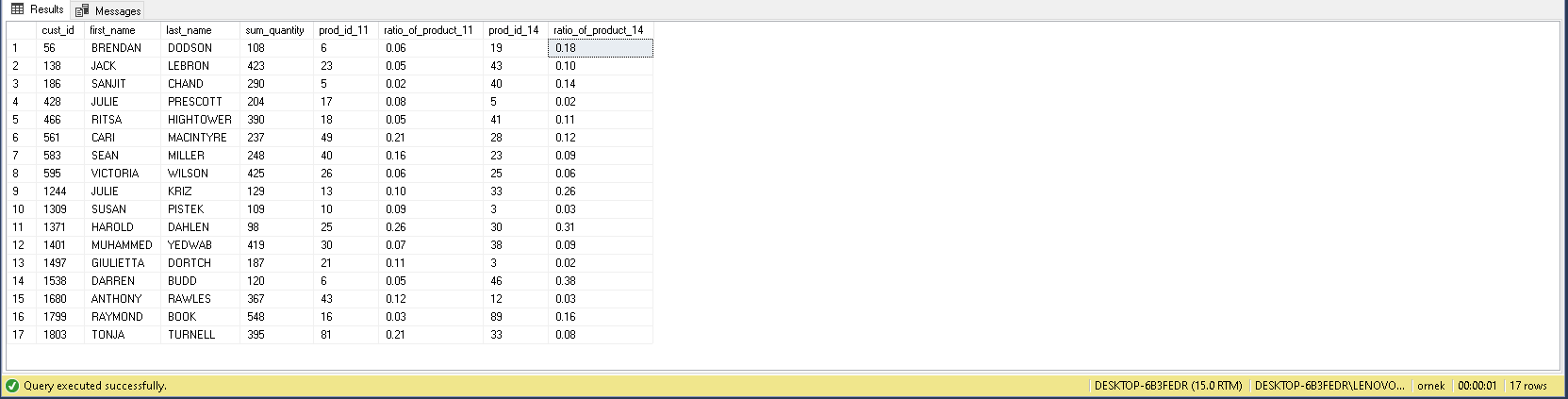
T1

INNER JOIN T2 ON T1.cust\_id = T2.cust\_id

INNER JOIN T3 ON T3.cust\_id = T2.cust\_id

ORDER BY

T1.cust\_id



**Customer Segmentation**

**Categorize customers based on their frequency of visits. The following steps will guide you. If you want, you can track your own way.**

/\* 1. Create a “view” that keeps visit logs of customers on a monthly basis. (For each log, three field is kept: Cust\_id, Year, Month) \*/

GO

;CREATE VIEW vw\_visit\_logs AS(

SELECT

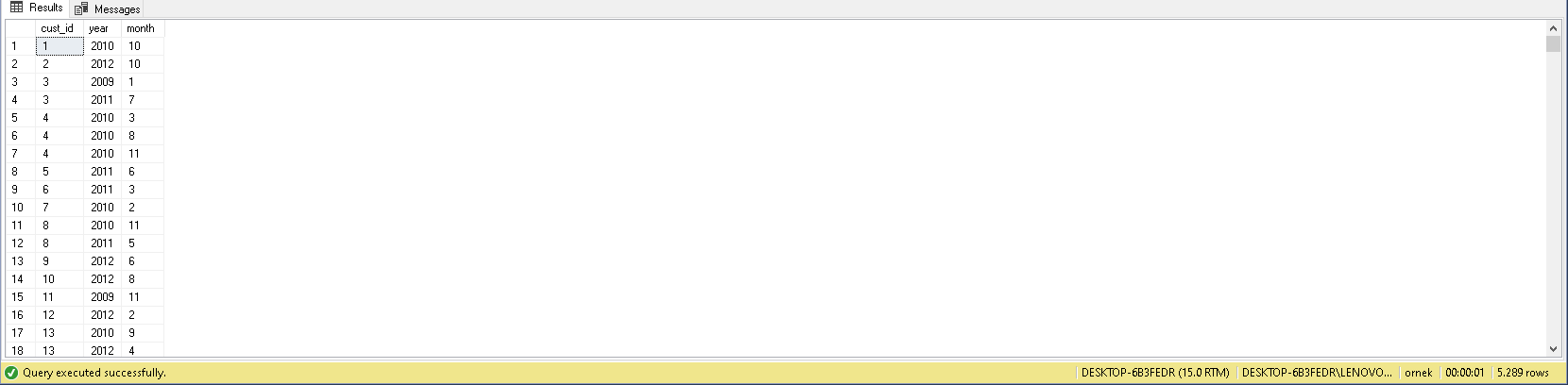
DISTINCT cust\_id,

YEAR(order\_date) AS 'year',

MONTH(order\_date) AS 'month'

FROM

[order].order\_table)



/\* 2. Create a “view” that keeps the number of monthly visits by users. (Show separately all months from the beginning business) \*/

CREATE VIEW vw\_number\_of\_visits AS(

SELECT

cust\_id,

[2009\_January], [2009\_February], [2009\_March], [2009\_April], [2009\_May], [2009\_June], [2009\_July],[2009\_August], [2009\_September], [2009\_October], [2009\_November], [2009\_December],

[2010\_January], [2010\_February], [2010\_March], [2010\_April], [2010\_May], [2010\_June], [2010\_July],[2010\_August], [2010\_September], [2010\_October], [2010\_November], [2010\_December],

[2011\_January], [2011\_February], [2011\_March], [2011\_April], [2011\_May], [2011\_June], [2011\_July],[2011\_August], [2011\_September], [2011\_October], [2011\_November], [2011\_December],

[2012\_January], [2012\_February], [2012\_March], [2012\_April], [2012\_May], [2012\_June], [2012\_July],[2012\_August], [2012\_September], [2012\_October], [2012\_November], [2012\_December]

FROM (

SELECT

cust\_id,

CONCAT(YEAR(order\_date), '\_', DATENAME(MONTH, order\_date)) AS YearMonth

FROM

[order].order\_table

) AS subquery

PIVOT (

COUNT(YearMonth)

FOR YearMonth IN (

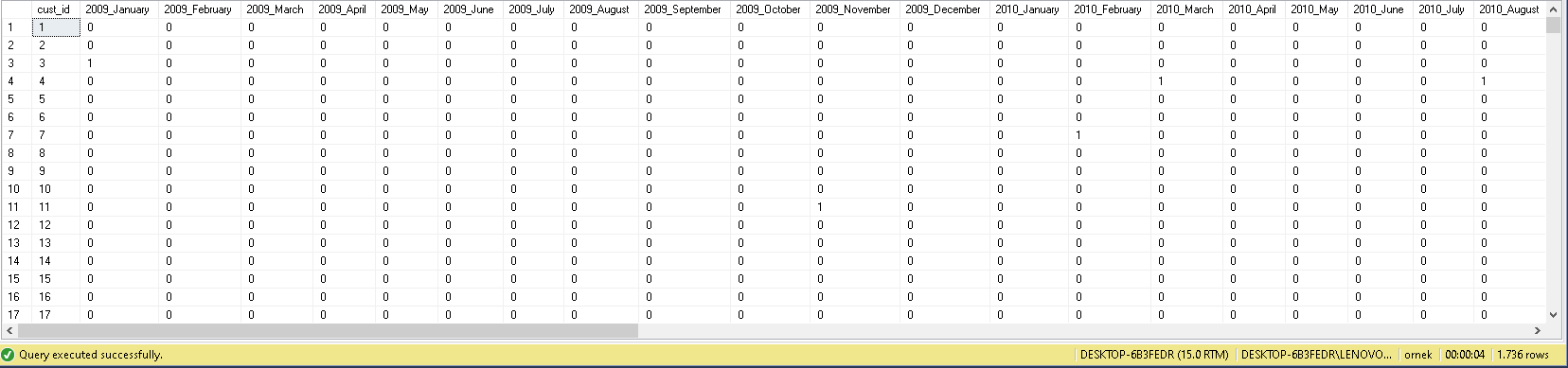
[2009\_January], [2009\_February], [2009\_March], [2009\_April], [2009\_May], [2009\_June], [2009\_July],[2009\_August], [2009\_September], [2009\_October], [2009\_November], [2009\_December],

[2010\_January], [2010\_February], [2010\_March], [2010\_April], [2010\_May], [2010\_June], [2010\_July],[2010\_August], [2010\_September], [2010\_October], [2010\_November], [2010\_December],

[2011\_January], [2011\_February], [2011\_March], [2011\_April], [2011\_May], [2011\_June], [2011\_July],[2011\_August], [2011\_September], [2011\_October], [2011\_November], [2011\_December],

[2012\_January], [2012\_February], [2012\_March], [2012\_April], [2012\_May], [2012\_June], [2012\_July],[2012\_August], [2012\_September], [2012\_October], [2012\_November], [2012\_December]

)

) AS pivot\_table);

/\* 3. For each visit of customers, create the next month of the visit as a separate column. \*/

SELECT

DISTINCT cust\_id,

order\_date AS [1\_visit],

LEAD(order\_date) OVER(PARTITION BY cust\_id ORDER BY order\_date) [2\_visit],

LEAD(order\_date, 2) OVER(PARTITION BY cust\_id ORDER BY order\_date) [3\_visit],

LEAD(order\_date,3) OVER(PARTITION BY cust\_id ORDER BY order\_date) [4\_visit],

LEAD(order\_date, 4) OVER(PARTITION BY cust\_id ORDER BY order\_date) [5\_visit],

LEAD(order\_date, 5) OVER(PARTITION BY cust\_id ORDER BY order\_date) [6\_visit],

LEAD(order\_date, 6) OVER(PARTITION BY cust\_id ORDER BY order\_date) [7\_visit],

LEAD(order\_date, 7) OVER(PARTITION BY cust\_id ORDER BY order\_date) [8\_visit],

LEAD(order\_date, 8) OVER(PARTITION BY cust\_id ORDER BY order\_date) [9\_visit],

LEAD(order\_date, 9) OVER(PARTITION BY cust\_id ORDER BY order\_date) [10\_visit],

LEAD(order\_date, 10) OVER(PARTITION BY cust\_id ORDER BY order\_date) [11\_visit],

LEAD(order\_date, 11) OVER(PARTITION BY cust\_id ORDER BY order\_date) [12\_visit],

LEAD(order\_date, 12) OVER(PARTITION BY cust\_id ORDER BY order\_date) [13\_visit],

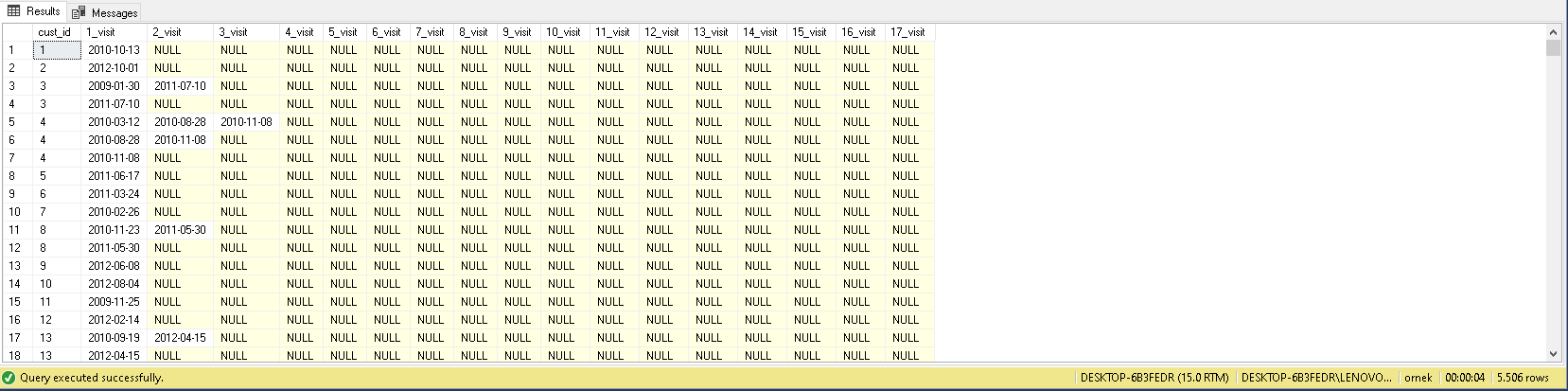
LEAD(order\_date, 13) OVER(PARTITION BY cust\_id ORDER BY order\_date) [14\_visit],

LEAD(order\_date, 14) OVER(PARTITION BY cust\_id ORDER BY order\_date) [15\_visit],

LEAD(order\_date, 15) OVER(PARTITION BY cust\_id ORDER BY order\_date) [16\_visit],

LEAD(order\_date, 16) OVER(PARTITION BY cust\_id ORDER BY order\_date) [17\_visit]

FROM

 [order].order\_table

/\* 4. Calculate the monthly time gap between two consecutive visits by each customer. \*/

WITH CTE AS (

SELECT

DISTINCT cust\_id,

0 AS [1\_visit],

DATEDIFF(MONTH ,order\_date, LEAD(order\_date) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [1\_gap],

DATEDIFF(MONTH , LEAD(order\_date) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 2) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [2\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 2) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 3) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [3\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 3) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 4) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [4\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 4) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 5) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [5\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 5) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 6) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [6\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 6) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 7) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [7\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 7) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 8) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [8\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 8) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 9) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [9\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 9) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 10) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [10\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 10) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 11) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [11\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 11) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 12) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [12\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 12) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 13) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [13\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 13) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 14) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [14\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 14) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 15) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [15\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 15) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 16) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [16\_gap]

FROM

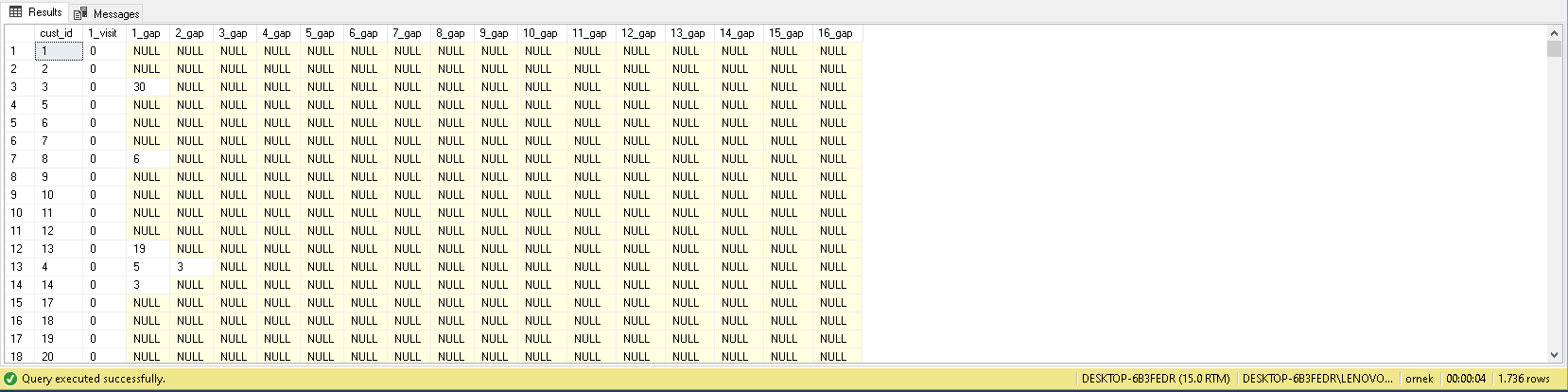
[order].order\_table

)

SELECT TOP 1 WITH TIES \*

FROM CTE

ORDER BY RANK() OVER (PARTITION BY cust\_id ORDER BY CASE WHEN [1\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [2\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [3\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [4\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [5\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [6\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [7\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [8\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [9\_gap] IS NULL THEN 1 ELSE 0 END +CASE WHEN [10\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [11\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [12\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [13\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [14\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [15\_gap] IS NULL THEN 1 ELSE 0 END + CASE WHEN [16\_gap] IS NULL THEN 1 ELSE 0 END + 0) ASC;



/\* 5. Categorise customers using average time gaps. Choose the most fitted labeling model for you. \*/

--create a table with 'total\_gaps\_non\_null' name

WITH CTE AS (

SELECT

DISTINCT cust\_id,

0 AS [1\_visit],

DATEDIFF(MONTH ,order\_date, LEAD(order\_date) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [1\_gap],

DATEDIFF(MONTH , LEAD(order\_date) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 2) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [2\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 2) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 3) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [3\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 3) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 4) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [4\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 4) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 5) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [5\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 5) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 6) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [6\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 6) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 7) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [7\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 7) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 8) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [8\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 8) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 9) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [9\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 9) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 10) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [10\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 10) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 11) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [11\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 11) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 12) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [12\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 12) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 13) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [13\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 13) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 14) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [14\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 14) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 15) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [15\_gap],

DATEDIFF(MONTH , LEAD(order\_date, 15) OVER(PARTITION BY cust\_id ORDER BY order\_date), LEAD(order\_date, 16) OVER(PARTITION BY cust\_id ORDER BY order\_date)) AS [16\_gap]

FROM

[order].order\_table

)

SELECT TOP 1 WITH TIES \*,

COALESCE([1\_gap], 0) + COALESCE([2\_gap], 0) + COALESCE([3\_gap], 0) + COALESCE([3\_gap], 0) + COALESCE([4\_gap], 0) + COALESCE([5\_gap], 0) + COALESCE([6\_gap], 0) + COALESCE([7\_gap], 0) + COALESCE([8\_gap], 0) + COALESCE([9\_gap], 0) + COALESCE([10\_gap], 0) + COALESCE([11\_gap], 0) + COALESCE([12\_gap], 0) + COALESCE([13\_gap], 0) + COALESCE([14\_gap], 0) + COALESCE([15\_gap], 0) + COALESCE([16\_gap], 0) AS total\_gap,

CASE WHEN [1\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [2\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [3\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [4\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [5\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [6\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [7\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [8\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [9\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [10\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [11\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [12\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [13\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [14\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [15\_gap] IS NOT NULL THEN 1 ELSE 0 END +

CASE WHEN [16\_gap] IS NOT NULL THEN 1 ELSE 0 END AS non\_null\_count

INTO total\_gaps\_non\_null

FROM CTE

ORDER BY RANK() OVER (PARTITION BY cust\_id ORDER BY CASE WHEN [1\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [2\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [3\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [4\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [5\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [6\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [7\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [8\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [9\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [10\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [11\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [12\_gap] IS NULL THEN 1 ELSE 0 END +

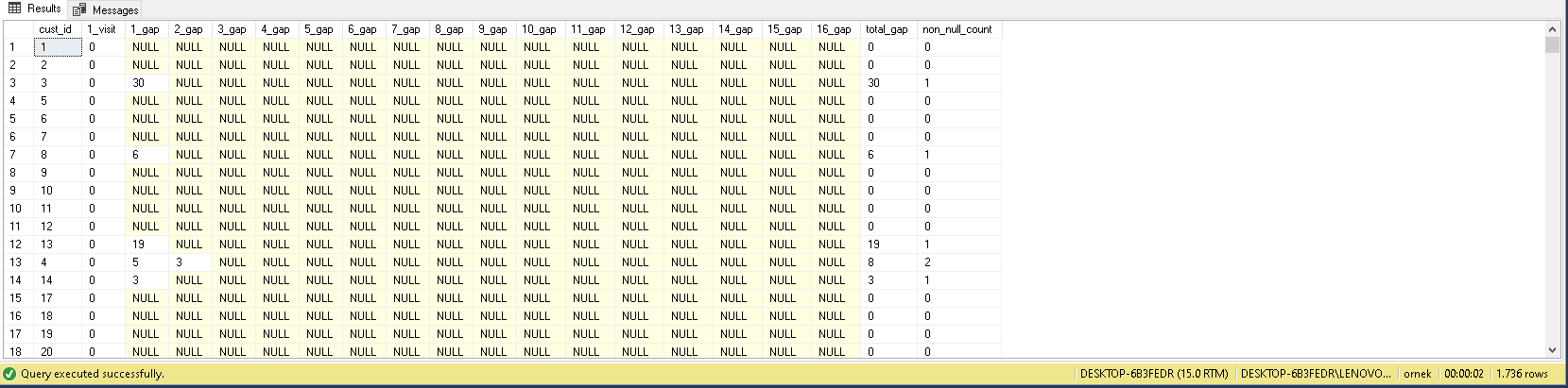
CASE WHEN [13\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [14\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [15\_gap] IS NULL THEN 1 ELSE 0 END +

CASE WHEN [16\_gap] IS NULL THEN 1 ELSE 0 END +

1. ASC

SELECT

cust\_id,

CASE WHEN avg\_gap = 0 THEN 'Churn'

WHEN avg\_gap = 1 THEN 'Regular'

WHEN avg\_gap BETWEEN 2 AND 10 THEN 'Very Good'

WHEN avg\_gap BETWEEN 11 AND 20 THEN 'Good'

WHEN avg\_gap BETWEEN 21 AND 30 THEN 'Medium'

WHEN avg\_gap BETWEEN 31 AND 40 THEN 'Bad'

ELSE 'Very Bad' END AS category\_visitors

FROM

(

SELECT

\*,

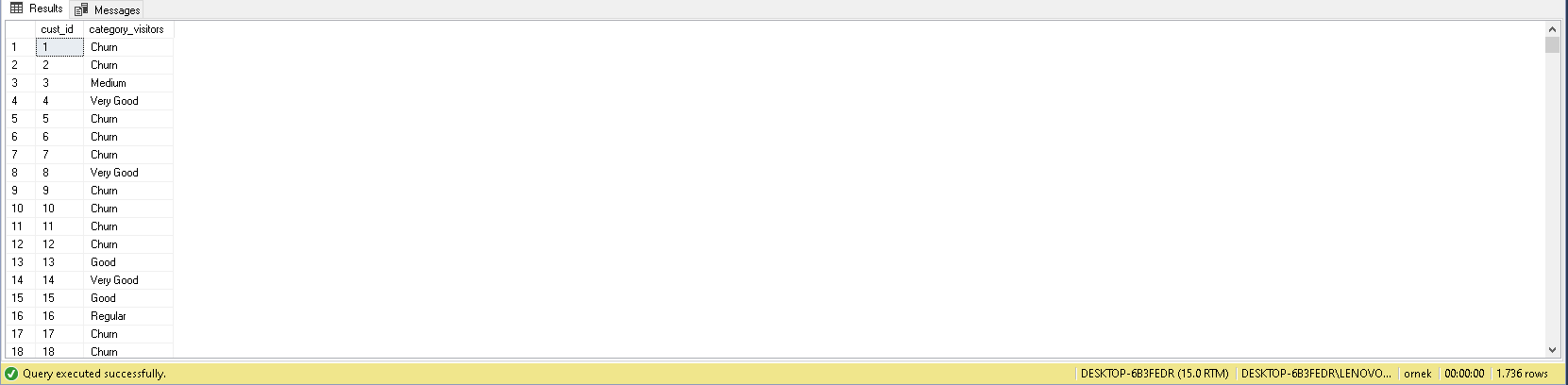
total\_gap / CASE WHEN non\_null\_count = 0 THEN 1 ELSE non\_null\_count END avg\_gap

FROM

total\_gaps\_non\_null) AS subquery

ORDER BY

cust\_id



**Month-Wise Retention Rate**

**Find month-by-month customer retention rate*i* since the start of the business.**