Target Customer List for Horizon Bank

Prepared by: North Star Consultants (GROUP 1)

Date: October 20, 2023

Submitted by:

Yian Chen

Sean Adam Kagugube

Emmanuella Acheampong

Susan Arzapalo

Table of Contents:

- **Introduction**
  - About North Star Consultants
    - *Mission Statement*
    - *Vision Statement*
    - *Our Services*
  - Why Invest in us?
  - Customer Background
  - Value Proposition
- **Phase 2: Target Customer List Development**
  - Minimum Deposit Qualification Criteria for New Product
    - Balance Criteria
    - Age Criteria
    - Loan Criteria
  - Justifications for the criteria used
  - Further Considerations/Analysis
  - Our Solution
- **Reflection**

1. Introduction

North Star Consultants

North Star Consultants is a leading consultancy dedicated to designing and implementing cutting-edge Database Management Systems (DBMS) for businesses across various industries. With a team of highly skilled database experts and a passion for data optimization, we specialize in helping financial institutions harness the full potential of their data resources.

Mission Statement:

"Our mission is to offer comprehensive DBMS solutions that enable financial institutions to unlock the true value of their data assets and ensure that our clients thrive in a rapidly evolving financial landscape."

Vision Statement:

"To be the trusted partner of choice for financial institutions seeking excellence in optimizing data storage, retrieval, and analysis, ensuring our clients stay ahead in today's data-driven world".

Our Services:

- **Data Quality Assurance**: We ensure your data is accurate, consistent, and reliable, eliminating errors and inaccuracies.
- **Data Transformation**: We convert raw data into actionable insights, seamlessly integrating data from multiple sources.
- **Insights for Success**: We provide client-centric insights to empower strategic planning and drive success.
- **Engaging Reporting**: Our visualizations and reports offer clarity and actionable recommendations.

Why Invest in us?

We have a history of over 100+ successful database implementations and satisfied clients in our 2 years of existence. Our solutions are high in demand and our expertise and commitment to excellence ensure investor confidence in our ability to execute and deliver results.

Customer Background

Horizon Bank is a prominent bank with over 50 branches across the United States. The bank wants to launch a new loan product for its loyal customers (more than 2 years with the bank). In addition, the bank is grappling with soaring marketing expenditures and an alarming drain on resources due to inefficient marketing campaigns. Horizon Bank is in dire need of a database with insights for optimizing their marketing strategies targeting the right customers, reducing expenses, and maximizing the effectiveness of their campaigns to maintain a competitive edge in the financial sector.

Value Proposition

Phase_1: To Build a relational database for managing the data set of the bank and leverage advanced data analytics and predictive modeling

Phase_2: To identify customers who qualify for the loan loyalty product for targeted marketing

Phase_3: To provide customer insights to enable the bank to develop a marketing strategy that minimizes expenses while delivering superior results.

Phase 2: Target Customer List Development

In this exciting second phase of our consultancy engagement with Horizon Bank, we are pleased to announce a pivotal project. Building upon the solid foundation of our newly implemented Database Management System (DBMS) in phase 1, Phase 2 focuses on creating a Target Customer List for a groundbreaking term deposit product set to be unveiled.

Horizon Bank's aspirations for this new product are underpinned by a robust set of requirements listed below.

Minimum Deposit Qualification Criteria for New Product:

1. **Balance Criteria:** Customers with a minimum monthly balance of $5,200 or more qualify.
2. **Age Criteria:** Customers aged between 18 and 65 are eligible.
3. **Loan Criteria:** Customers must have existing housing or personal loans to qualify.

Justifications for the criteria used:

The selected criteria for this customer qualification process serve multiple essential purposes. Firstly, the financial stability criterion ensures that customers have the means to commit to the new product, thus minimizing financial risk. The age range criterion offers a broad and diverse customer base, optimizing the product's reach. Identifying customers with existing loans, a significant financial relationship indicator, positions them as suitable candidates for additional products. The bank is specifically interested in customers with existing loans because it already has a record of the financial records of these customers and has established a long-lasting relationship with them.

One may ask, why would the bank want to put criteria in place for customers who want to

deposit their funds with the bank. The answer is, since the new term deposit product offers a high interest rate, the bank has put in place these requirements in order to have loyal customers enjoy this high interest rate offer.

By combining these criteria related to customer loyalty, financial capability, and unique preferences, we can now present a finely curated list of potential customers who not only qualify for the new product but are also highly likely to embrace it. This data-driven approach optimizes resource allocation and enhances the prospects of a successful product launch.

Further Considerations/Analysis

1. Priority Status: Customers fall into three priority categories: "High_Priority," "Low_Priority," and "Disqualified." The categorization of customers into high-priority (monthly balance over $20,000), low-priority (balances between $5,200 and $20,000) and disqualified (below $5,200) segments tailors the product to customers' distinct financial profiles and this will help the bank to know which customers to prioritize during its marketing campaign.

2. Loan and Age Data Analysis: The average age of customers with existing loans is considered. Analyzing the average age of customers with existing loans (who are the bank's target) provides insights into their life stage and financial commitments, helping predict their inclination toward longer-term deposit options. This data aids in tailoring the new term deposit product to better meet their needs and encourages longer-term deposits.

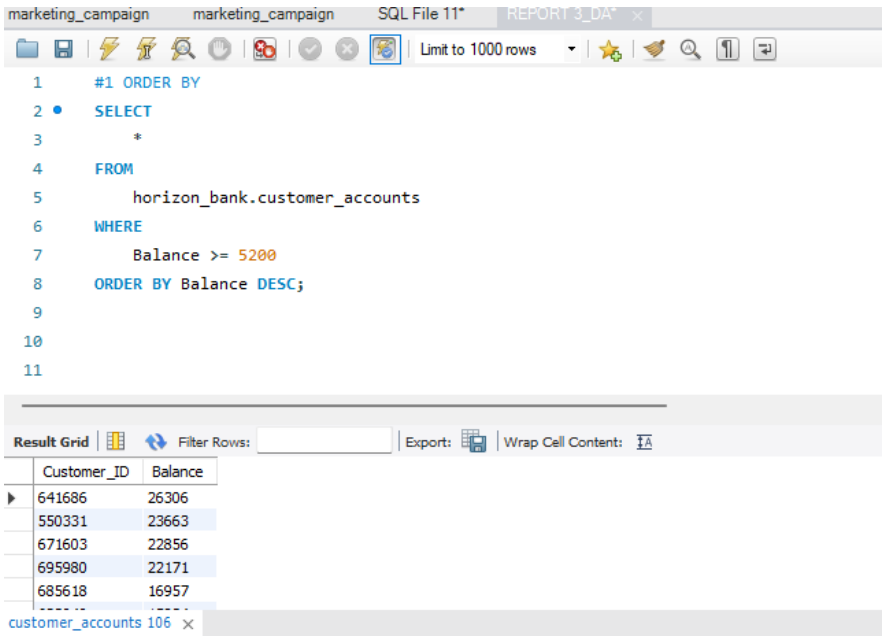
Our Solution

Our approach centers on data-driven precision. To meet these requirements, our team has conducted a comprehensive analysis, employing 28 SQL queries that carefully select individuals who align with the desired criteria.

Our approach centers on data-driven precision. To meet these requirements, our team has conducted a comprehensive analysis, employing 28 SQL queries that carefully select individuals who align with the desired criteria.

Query 1: ORDER BY

Objective: The objective of this query is to assess the balance distribution of the bank's customers with a monthly balance greater than or equal to $5,200. The aim is to generate a pool of customers who meet the first requirement of having a minimum balance of $5,200.



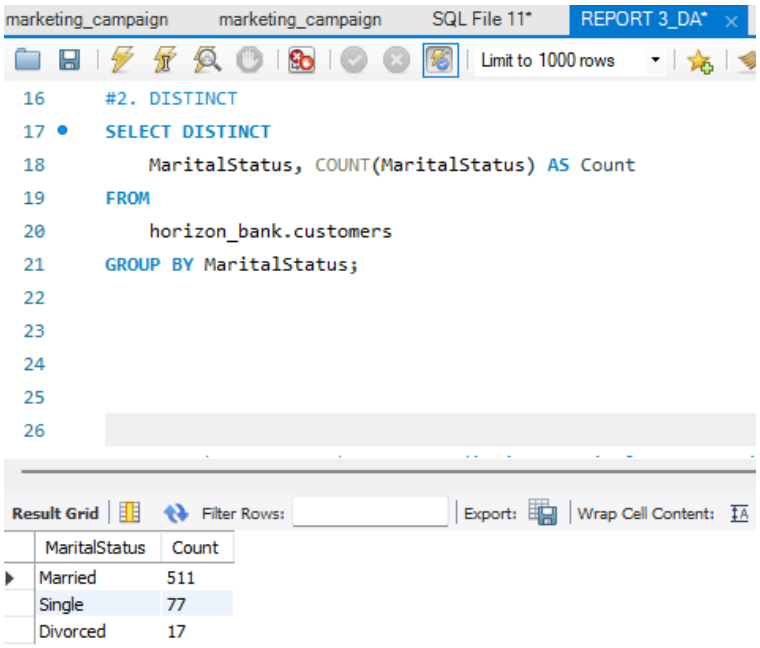
Output Report:

The query filters the dataset to include only those customers with a monthly balance greater than or equal to $5,200 and then **orders** the balances in descending order. The ordered output reveals a balance distribution ranging between $5,200 and $26,306.

From the output, 47 rows were returned out of a total of 605 customers.

Query 2: DISTINCT

Objective: To analyze the distinct marital statuses of the bank's customers. Chances are that, married couple may want to have long term investments to support the educational needs of their children. It could also be argued that divorced or single customers may have low commitments and would therefore want to invest their surplus funds for a long term.



Output Report:

The query identifies the distinct marital segments within the customer dataset and provides a count of customers within each segment. The output revealed that the majority of the bank's customers are married. In light of this, we sought to get the proportion of married customers who met the first requirement (maintained a minimum balance of $5.200 or more). The findings as revealed in the output below suggested that 41 out of the 47 customers who met the first requirement are married. Please see rows returned (highlighted in blue) from the below screenshot.

marketing_campaign | marketing_campaign | SQL File 11* | REPORT 3_DA*

Limit to 1000 rows

```
#1 ORDER BY
SELECT
    *
FROM
    horizon_bank.customer_accounts
    LEFT JOIN horizon_bank.customers USING(Customer_ID)
WHERE
    Balance >= 5200
    AND MaritalStatus = "Married"
ORDER BY Balance DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_ID	Balance	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
550331	23663	Married	Colten	Dean	US	33	housemaid	tertiary
695980	22171	Married	Reign	Preston	US	29	admin.	secondary
685618	16957	Married	Cason	Lawson	US	38	management	tertiary
623940	15834	Married	Krew	Hull	US	70	retired	tertiary
593388	15459	Married	Mckenna	Knox	US	29	management	tertiary

Result 28

Output

Action Output

#	Time	Action	Message
27	11:59:46	SELECT * FROM horizon_bank.customer_accounts WHERE Balance >= 5200 ORDER BY Bala...	47 row(s) returned
28	12:15:28	SELECT * FROM horizon_bank.customer_accounts LEFT JOIN horizon_bank.customers USING(	41 row(s) returned

Query 3: AND

Objective: Now we moved a step further to identify the customers who meet the second requirement of being 20 years or older in addition to the first minimum balance requirement. The age factor can guide the bank in knowing the likely duration a customer may want to keep funds in term deposit. For example, aged customers may want to liquidate their funds soon to cater for their retirement needs compared to younger customers.

```
#3 AND
SELECT
    CA.*, Age
FROM
    horizon_bank.customer_accounts CA
        LEFT JOIN
    horizon_bank.customers C USING (Customer_ID)
WHERE
    Balance >= 5200
        AND Age BETWEEN 18 AND 65
ORDER BY Age DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

Customer_ID	Balance	Age
641686	26306	54
541335	7317	50
548540	12519	50
526302	8545	49
549625	6269	46

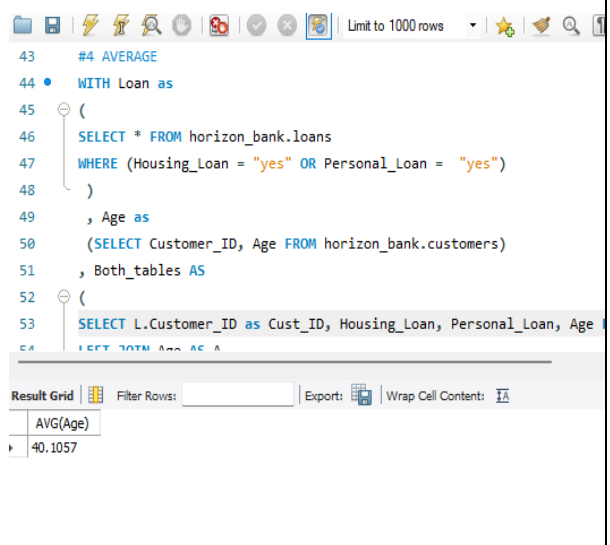
Output Report:

This query selects customers who satisfy two conditions: a monthly balance greater than or equal to $5,200 and an age between 18 and 65. The results are ordered by balance in descending order.

7 out of the 47 customers were dropped as they were either less than 18 years old or more than 65 years old, leaving just 40 customers to target.

Query 4: AVERAGE

Objective: Since it is also a requirement for a customer to have either a Personal or Housing Loan, the next step seeks to calculate the average age of customers with housing or personal loans.

QUERY: #4 AVERAGE	OUTPUT
WITH Loan as (SELECT * FROM horizon_bank.loans WHERE (Housing_Loan = "yes" OR Personal_Loan = "yes")) , Age as (SELECT Customer_ID, Age FROM horizon_bank.customers) , Both_tables AS (SELECT L.Customer_ID as Cust_ID, Housing_Loan, Personal_Loan, Age FROM Loan AS L LEFT JOIN Age AS A	

ON L.Customer_ID = A.Customer_ID) SELECT AVG(Age) FROM Both_tables;	

Output Report:

This query calculates the average age of customers who have either a housing loan or a personal loan. The first CTE identifies customers who have either loans from the loan table. The second CTE selects the ages of the customers from the customers table. The third CTE selects the customers who have either loans from the first CTE and joins it to their respective ages from the second CTE. Then finally, the query after the 3 CTEs selects the ages and performs an average function to get the average age.

The result indicates that the average age of the bank's customers who have either loans is 40 years irrespective of whether they have met the first two requirements. That is, it only focuses on finding the average age of customers who have either of the loans.

Query: 5 BETWEEN

Objective: The objective is to identify customers aged between 18 and 65 for the second requirement.

This query selects customers whose ages fall within the range of 18 to 65 years. The results are ordered by balance in descending order.

```
73  ----------------------------------------------------
74  #5 BETWEEN
75  SELECT
76      CA.*, Age
77  FROM
78      horizon_bank.customer_accounts CA
79          LEFT JOIN
80      horizon_bank.customers C USING (Customer_ID)
81  WHERE
82      Age BETWEEN 18 AND 65
```

Result Grid | Filter Rows: | Export: | Wrap Cell Con

Customer_ID	Balance	Age
526022	1840	65
766146	493	65
550705	43	64
584326	768	64
701091	846	64

Output Report:

In this result above, we realized that the oldest customer is 65 years old and the youngest customer is 20 years old when ordered by age. So we asked, is age correlated with customers account balance? From the result, age is not correlated with account balance. This is because, we can see a customer who is 64 years but has a balance as low as $43. Again, when this same output is ordered by account balance instead of age (from the screenshot below), the customer with the highest balance is less than 65 years. Still, this is interesting because one would have inferred that an old customer would be expected to have a lower balance because they may not be engaged in an active income generating job.

```sql
#5 BETWEEN
SELECT
    CA.*, Age
FROM
    horizon_bank.customer_accounts CA
        LEFT JOIN
    horizon_bank.customers C USING (Customer_ID)
WHERE
    Age BETWEEN 18 AND 65
ORDER BY Balance DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cel

Customer_ID	Balance	Age
641686	26306	54
550331	23663	33
671603	22856	37
695980	22171	29
685618	16957	38

Query 6: CASE

Objective: Next, we aimed to categorize customers into different priority groups based on their account balances. Categorizing customers by priority status allows the bank to tailor its product offering and marketing strategies to different customer segments. This approach optimizes resource allocation and enhances the chances of success.

The query below classifies customers into three priority categories: "High_Priority" for balances over $20,000, "Low_Priority" for balances between $5,200 and $20,000, and "Disqualified" for balances below $5,200.

```sql
#6 CASE
SELECT
    Customer_ID,
    Balance,
    CASE
        WHEN Balance > 20000 THEN 'High_Priority'
        WHEN Balance BETWEEN 5200 AND 20000 THEN 'Low_Priority'
        ELSE 'Disqualified'
    END Priority_Status
FROM
    horizon_bank.customer_accounts;
```

Customer_ID	Balance	Priority_Status
549625	6269	Low_Priority
550162	252	Disqualified
550331	23663	High_Priority
550634	81	Disqualified
550705	43	Disqualified

Output Report:

The output below shows the distribution of customers as high priority, low_priority and disqualified customers based on their account balances. Majority of the customers were disqualified as they maintained a low balance.

Query 7: COUNT

Objective: Having classified customers into various priority statuses, we were interested in knowing the number of customers in each priority status

The query below builds on the CASE query above by turning it into a CTE and then writing a select statement from the CTE that counts the number of customers in each priority status category, providing insight into the distribution of customers across these categories.

```sql
#7 COUNT
WITH P_status AS
(
SELECT Customer_ID, Balance, CASE WHEN Balance > 20000 THEN "High_Priority"
WHEN Balance BETWEEN 5200 AND 20000 THEN "Low_Priority"
ELSE "Disqualified"
END Priority_Status
FROM horizon_bank.customer_accounts
)
SELECT Priority_Status, COUNT(Priority_Status) AS Status_Count FROM P_status
GROUP BY Priority_Status;
```

Priority_Status	Status_Count
Disqualified	558
Low_Priority	43
High_Priority	4

Output Report:

The result reveals that the majority of the bank's customers fall within the "disqualified" category and only a few fall within the high priority category. However, in total, without factoring age, 47 customers (low and high priority customers) can be targeted by the bank for the new product. This affirms our result from the "order by" query above.

Query 8: GROUP BY

Objective: The objective here is to get an overview of the number of people with or without loans. Grouping and counting loans by type helps the bank assess its loan portfolio and better understand the distribution of different loan products among its customers.

```
#8 GROUP BY
SELECT
    Housing_Loan, Personal_Loan, COUNT(*) AS Count_of_Loans
FROM
    Horizon_Bank.loans
GROUP BY Housing_Loan , Personal_Loan;

-----------------------------------------------------------------
#9 HAVING
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Housing_Loan	Personal_Loan	Count_of_Loans
yes	no	316
yes	yes	56
no	no	217
no	yes	16

Output Report:

From the above result, we realized that the majority of the bank's customers have housing loans relative to personal loans. Also, the majority of the customers do not have any of the loans. The bank may therefore want to take advantage of this insight to increase its loan base so as to ensure that most of the customers will have loans and subsequently qualify for the new term deposit which comes with a prerequisite of having a loan.

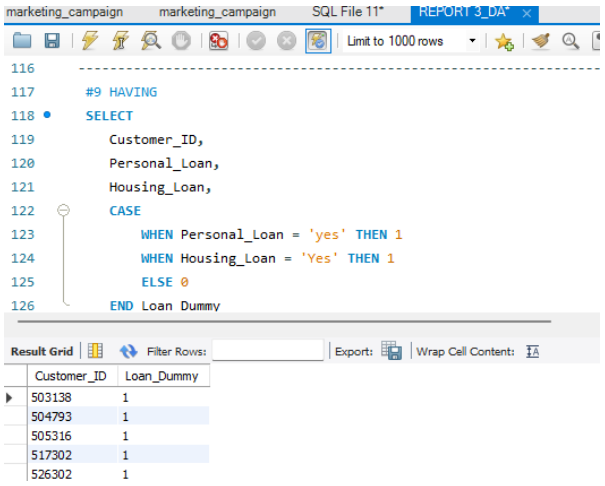
Query 9: HAVING

Objective: Here, we aim to identify customers who have both personal and housing loans. Identifying customers with both types of loans can provide insights into customers' financial commitments. Such customers might be interested in additional financial products or services.

The query below aims to find customers who have both personal and housing loans. Because the responses to whether a customer has a loan or not is categorized as “YES” or “NO”, the query first creates a "Loan_Dummy" value for each customer, where 1 indicates the presence of both loan types and 0 indicates otherwise. It then groups these responses by Customer_ID having the sum of loans greater than zero.

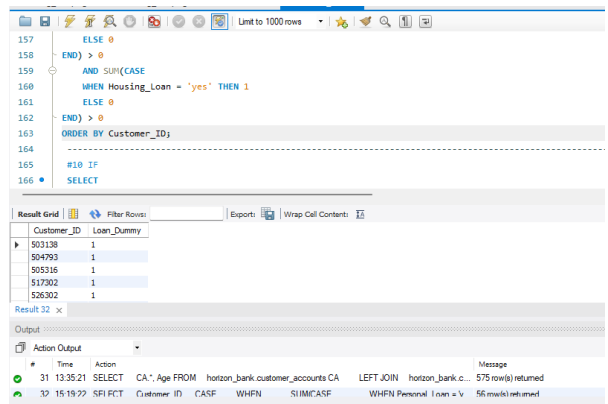
#9 HAVING

```
 SELECT
   Customer_ID,
   Personal_Loan,
   Housing_Loan,
   CASE
      WHEN Personal_Loan = 'yes' THEN 1
      WHEN Housing_Loan = 'Yes' THEN 1
      ELSE 0
   END Loan_Dummy
FROM
   Horizon_Bank.loans
GROUP BY Customer_ID
HAVING COUNT(Personal_Loan) > 0
   AND COUNT(Housing_Loan) > 0
ORDER BY Customer_ID;

SELECT
   Customer_ID,
   CASE
      WHEN
         SUM(CASE
```



Continued …

```
            WHEN Personal_Loan = 'yes'
THEN 1
            ELSE 0
        END) > 0
            AND SUM(CASE
            WHEN Housing_Loan = 'yes'
THEN 1
            ELSE 0
        END) > 0
      THEN
        1
      ELSE 0
   END AS Loan_Dummy
FROM
   Horizon_Bank.loans
GROUP BY Customer_ID
HAVING SUM(CASE
   WHEN Personal_Loan = 'yes' THEN 1
   ELSE 0
END) > 0
   AND SUM(CASE
   WHEN Housing_Loan = 'yes' THEN 1
   ELSE 0
END) > 0
ORDER BY Customer_ID;
```



Output Report:

From the output (row returned), we realized that 56 people have both Personal loan and Housing Loan.

Query 10: IF

Objective: This section categorizes customers based on subscription status. This is because the bank may want to reach out to customers who qualify for the new product and from the dataset. Since some customers ticked no when asked to be contacted by the bank through a phone call, we aimed to generate a list of customers, separating them by their call options.

The query below categorizes customers into two groups: "Can_Call" for those with a subscription result of "YES" and "Do_Not_Call" for those with no.

```
-----------------------------------------
#10 IF
SELECT
    *,
    IF(Subscribe_Result = 'YES',
        'Can_Call',
        'Do_Not_Call') AS Call_Option
FROM
    horizon_bank.subscriptions;

-----------------------------------------
```

Result Grid | Filter Rows: | Export: | Wra

Customer_ID	Subscribe_Result	Call_Option
502654	yes	Can_Call
503138	no	Do_Not_Call
503833	no	Do_Not_Call
504127	no	Do_Not_Call
504793	no	Do_Not_Call

Output Report:

From the result, the majority of the customers requested not to be reached by phone. This indicates that the bank has to find alternative means of reaching out to the customers who qualify for the term deposit product but opted out of call subscription.

Query 11: IN

Objective: This query builds on query 6 and 7 (which categorized customers into various priority statuses) and goes on to identify who the customers in the various categories are. This is to enable the bank to call the customers who qual

marketing_campaign | marketing_campaign | SQL File 11* | REPORT 3_DA*

Limit to 1000 rows

```
#11 IN
WITH P_status AS
(
SELECT Customer_ID, Balance, CASE WHEN Balance > 20000 THEN "High_Priority"
WHEN Balance BETWEEN 5200 AND 20000 THEN "Low_Priority"
ELSE "Disqualified"
END Priority_Status
FROM horizon_bank.customer_accounts
)
SELECT Customer_ID, First_Name, Last_Name, Priority_Status FROM P_status
LEFT JOIN horizon bank.Customers USING(Customer ID);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_ID	First_Name	Last_Name	Priority_Status
502654	Karsyn	Ballard	Disqualified
503138	Riley	Medina	Disqualified
503833	Haisley	Henson	Disqualified
504127	Paris	Conner	Disqualified
504793	Ailani	Weiss	Disqualified

Output Report:

The output from this query is similar to the query on “COUNT” but this query rather than count the number of customers, provides the names of the customers in the various categories by combining the balance and customer tables.

Query 12: INNER JOIN

Objective: This query seeks to combine customer and subscription data to identify only customers who have subscribed "YES" for call options.

The query below performs an inner join between customer and subscription data to identify customers who have subscribed.

```
#12 INNER JOIN
SELECT
    *
FROM
    horizon_bank.subscriptions
        INNER JOIN
    horizon_bank.Customers USING (Customer_ID)
WHERE
    Subscribe_Result = 'YES';

--------------------------------------------------------------------------------------------
```

Customer_ID	Subscribe_Result	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
502654	yes	Married	Karsyn	Ballard	US	31	services	primary
507367	yes	Married	Leo	Velazquez	US	33	management	secondary
507525	yes	Married	Jaliyah	Rangel	US	61	admin.	other
508337	yes	Married	Matias	Rollins	US	37	management	tertiary
509321	yes	Married	Anika	Gillespie	US	52	services	secondary

Output Report::

The result gives the bank a fair idea of the number of customers who they can call for marketing campaigns. Only 146 out of the 605 customers opted "YES" to call option.

Query 13: LEFT JOIN

Objective: Having identified the customers with loans from the loans table, this query seeks to combine customer data from the customers table with loan information to give a complete view of who those customers are..

The below query combines customer data with loan information through a left join. It identifies customers who have loans, either housing loans or personal loans.

marketing_campaign | marketing_campaign | SQL File 11* | REPORT 3_DA*

Limit to 1000 rows

```sql
#13 LEFT JOIN
SELECT
  Housing_Loan, Personal_Loan, C.*
FROM
  horizon_bank.loans
      LEFT JOIN
  horizon_bank.Customers AS C USING (Customer_ID)
WHERE
  (Housing_Loan = 'YES'
      OR Personal_Loan = 'YES');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Housing_Loan	Personal_Loan	Customer_ID	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
yes	no	502654	Married	Karsyn	Ballard	US	31	services	primary
yes	yes	503138	Married	Riley	Medina	US	33	services	secondary
yes	no	503833	Married	Haisley	Henson	US	35	management	tertiary
yes	yes	504793	Married	Ailani	Weiss	US	43	services	primary
yes	yes	505316	Married	David	Fuller	US	31	blue-collar	secondary

Output Report:

The result is a combination of the loans table and customers table to give details of the customers who have loans with the bank.

Query 14: LIKE

Objective: The aim is to identify customers whom the bank has been successful in contacting for marketing campaigns in the past. Identifying customers with successful contacts can provide insights into effective communication and customer engagement strategies.

The query below selects customers who have had previous contacts with the term "Success" in their outcome from the contacts table.

```
206
207     ----------------------------------------------------------------------------------
208      #14 LIKE
209      SELECT
210         *
211     FROM
212         horizon_bank.contacts
213     WHERE
214         Previous_Outcome LIKE '%Success%';
215
216      ----------------------------------------------------------------------------------
```

Customer_ID	Contact	Day_of_Month	Last_Contact_Month	Duration_Second	Campaign_Times	Past_Days	Previous_Contact	Previous_Outcome
502654	cellular	7	may	623	1	293	1	success
507525	cellular	27	jan	181	1	92	1	success
508337	cellular	16	jul	268	2	182	3	success
510318	cellular	12	feb	473	1	182	1	success
513786	cellular	30	jun	258	6	169	3	success

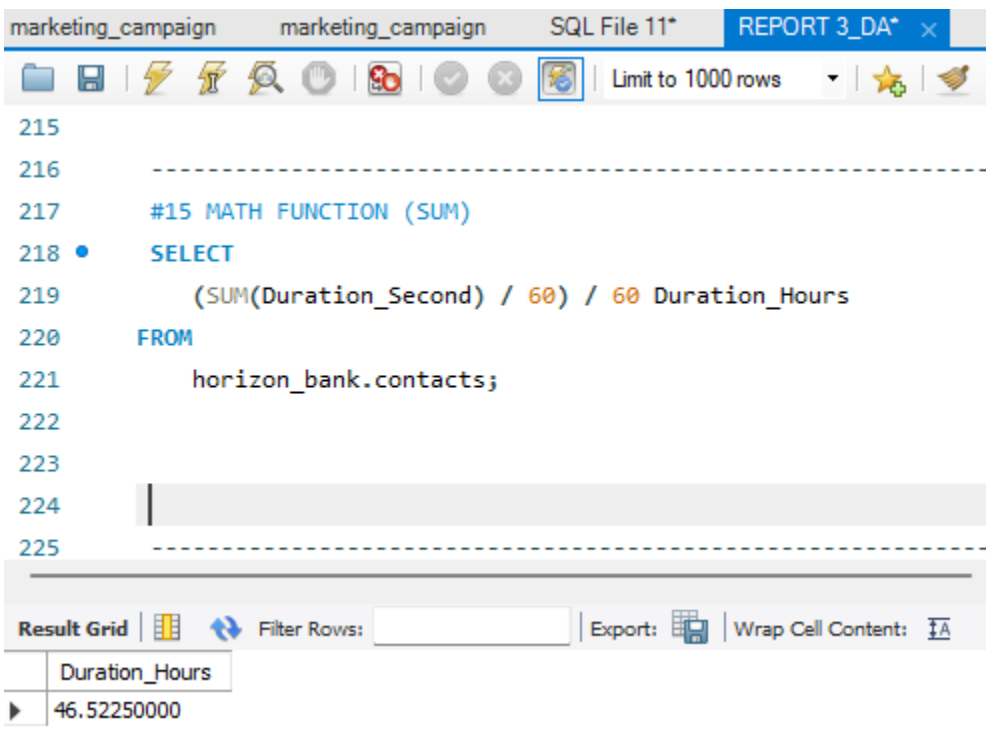
Output Report:

The result shows that only 127 customers out of 605 customers were contacted with success over the last year by the bank. This indicates that communication between the bank and its customers is not really effective.

Query 15: Math Function - SUM

Objective: Having established the number of customers who the bank contacted in the past successfully, we aim to understand the duration of contact time between the bank and its customers. Calculating the total duration of customer contacts provides insights into the volume of interactions and helps in resource allocation for customer support.

The below query calculates the total duration of customer contacts in hours by summing the "Duration_Second" field and dividing by 60 and for duration in minutes and further dividing by 60 for duration in hours.



Output Report:

We found that the bank only spent approximately 47 hours in contacting its customers over one year. It could be that there are other means by which the bank communicates with its customers.

However, this requires further investigation.

Query 16: MAX

Objective: The aim here is to find the oldest customer's age. Knowing the age of the oldest customer can help in understanding the longevity of customer relationships and tailoring services for different age groups.

The query below identifies the oldest customer's age within the dataset by using the MAX function.

```
224
225     ----------------------------------------
226     #16 MAX
227     SELECT
228        MAX(Age) AS Oldest
229    FROM
230        horizon_bank.customers;
231
232
233
234     ----------------------------------------
```

Oldest
83

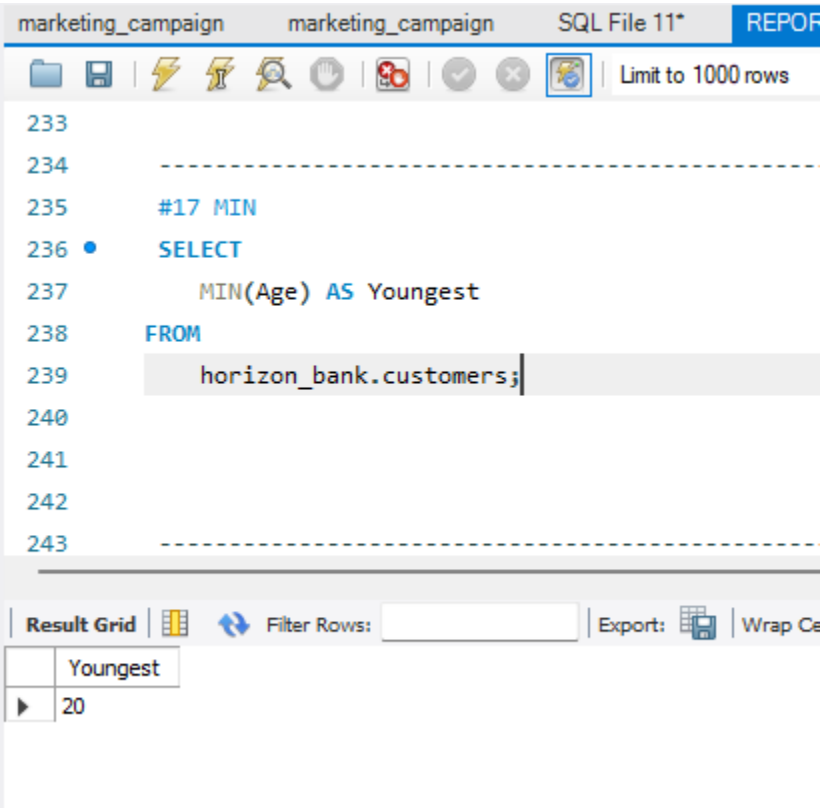
Output Report:

As seen in the results from the query, the oldest customers of the bank are aged 83 years old. However, the dataset doesn't give information about the duration of the relationship with its customers hence we cannot conclude if the customers in this age bracket have the longest relationship with the bank.

Query 17: MIN

Objective: Contrary to the above query which sought to find the age of the oldest customer, the objective here is to find the youngest customer's age. Understanding the age of the youngest customer provides insights into attracting and retaining a younger customer base.

The below query identifies the youngest customer's age within the dataset using the MIN function.



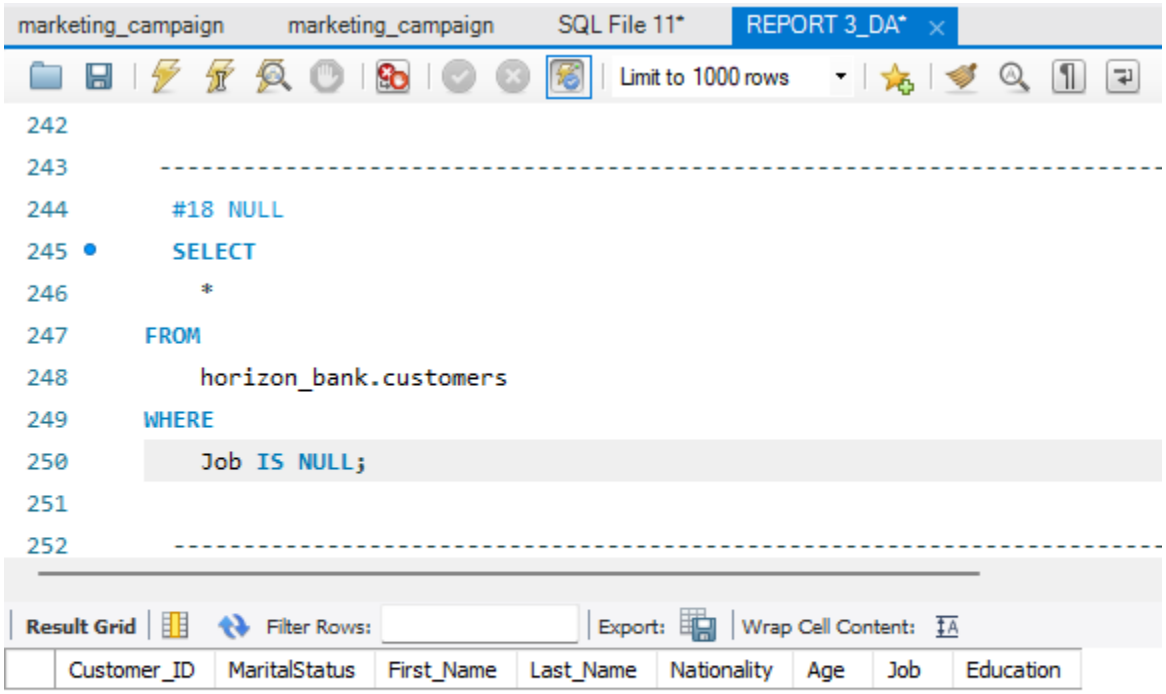
Output Report:

We found out that the youngest customer is 20 years old. This gives a fair change to all young customers to qualify for the new product as the qualifying age starts from 18 years.

Query 18: NULL

Objective: The aim here is to Identify customers with a missing job field. Identifying customers with missing job information is crucial for data completeness and improving customer profiles.

The query below selects customers whose "Job" field is NULL, indicating missing job information.



Output Report:

Evidently from the screenshot above with no output, we can conclude that all customers in the bank's database have the job field not being empty. There are no NULLS for all other fields in the database. This suggests complete customer profiles from which good insights can be drawn.

Query 19: OR

Objective: Unlike the query above that identifies customers having only one type of loan, this step identifies customers without loans.

The below query combines the customers table with the loans table and identifies customers without any loans.

```
#19 OR
SELECT
  Housing_Loan, Personal_Loan, C.*
FROM
  horizon_bank.loans
      LEFT JOIN
  horizon_bank.Customers AS C USING (Customer_ID)
WHERE
  (Housing_Loan = 'NO'
      OR Personal_Loan = 'No');
```

Housing_Loan	Personal_Loan	Customer_ID	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
yes	no	502654	Married	Karsyn	Ballard	US	31	services	primary
yes	no	503833	Married	Haisley	Henson	US	35	management	tertiary
no	no	504127	Married	Paris	Conner	US	35	management	tertiary
yes	no	505661	Married	Lilyana	Clarke	US	37	admin.	tertiary
no	no	505721	Married	Stetson	Gray	US	67	retired	other

Output Report:

The result of this query is quite intriguing. Initially, we anticipated that it would retrieve only customers without any loans. However, we discovered that the query did not only captured customers without any loans, as we expected, but also included customers who had either housing loans or personal loans. This outcome has taught us an important lesson in SQL query logic. When the housing loan status is set to 'no,' the OR function within SQL moves on to the next option, in this case, personal loans to look for customers without personal loans which at the same time result in a 'yes' for housing loans and a 'no' for personal loans in certain cases.

We can conclude that the best syntax to use here is the “AND” syntax which can be stated as WHERE Personal_loan = “NO” and Housing_Loan = “NO” to get customers without loans like the below screenshot shows.

marketing_campaign | marketing_campaign | SQL File 11* | REPORT 3_DA*

Limit to 1000 rows

```
256   SELECT
257     Housing_Loan, Personal_Loan, C.*
258 FROM
259     horizon_bank.loans
260         LEFT JOIN
261     horizon_bank.Customers AS C USING (Customer_ID)
262 WHERE
263     (Housing_Loan = 'NO'
264         AND Personal_Loan = 'No');
265 ----------------------------------------------------------------------------------------------
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Housing_Loan	Personal_Loan	Customer_ID	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
no	no	513786	Married	Julieta	Cisneros	US	21	student	secondary
no	no	513798	Married	Alden	Rodgers	US	33	blue-collar	secondary
no	no	516758	Married	Sophie	Watson	US	32	blue-collar	secondary
no	no	517511	Single	Abraham	Doyle	US	55	admin.	secondary
no	no	517812	Single	Maison	Friedman	US	28	management	tertiary

Result 37

Query 20: RIGHT JOIN

Objective: The objective here is to combine customer and subscription data to identify customers who have not subscribed to a call option. Understanding which customers have not subscribed can assist in targeting these customers with other communication offers and improving conversion rates.

The query below performs a right join between the customers table and subscriptions table, identifying customers who have not subscribed (Subscribe_Result = "NO").

marketing_campaign | marketing_campaign | SQL File 11* | REPORT 3_DA*

Limit to 1000 rows

```
        horizon_bank.Customers AS C USING (Customer_ID)
    WHERE
        (Housing_Loan = 'NO'
            OR Personal_Loan = 'No');
    ----------------------------------------------------------------------------------------
    #20 RIGHT JOIN
    SELECT * FROM horizon_bank.customers
    RIGHT JOIN horizon_bank.subscriptions USING(Customer_ID)
    WHERE Subscribe_Result = "NO";

    ----------------------------------------------------------------------------------------
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_ID	Subscribe_Result	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
503138	no	Married	Riley	Medina	US	33	services	secondary
503833	no	Married	Haisley	Henson	US	35	management	tertiary
504127	no	Married	Paris	Conner	US	35	management	tertiary
504793	no	Married	Ailani	Weiss	US	43	services	primary
505316	no	Married	David	Fuller	US	31	blue-collar	secondary

Output Report:

459 out of the 605 customers have “NO” as a subscription result. This is more than 50% of the bank’s customers and indicates a low volume of communication between the bank and its customers. This can influence customer churning for some of the customers.

Query 21: String Function - UPPER

Objective: The objective here is to create a list of customer full names in uppercase for data consistency and suitability in various reports and applications.

The query below generates a list of customer full names with uppercase letters using the CONCAT and UPPER syntax.

```
---------------------------------------------------------------------
#21 String Function(UPPER)
SELECT
    CONCAT(UPPER(First_Name), ' ', UPPER(Last_Name)) AS Full_Name
FROM
    horizon_bank.customers;

---------------------------------------------------------------------
#22 UNION
WITH P loans AS
```

Full_Name
KARSYN BALLARD
RILEY MEDINA
HAISLEY HENSON
PARIS CONNER
AILANI WEISS

Output Report:

The output is a list of customers' first names combined with their last names in upper case.

Query 22: UNION

Objective: The aim is to combine two subsets of loan data (personal and housing loans) to create a unified list of loans. This is to streamline loan management and analysis processes. Since we didn't have two tables with similar characteristics to perform a union, we resorted to creating CTEs as tables and creating a union out of them.

The query below creates 2 CTEs, one for personal loans and another for housing loans and combines two subsets of loan data through a union. It produces a unified list of loans ordered by customer ID.

```
#22 UNION
WITH P_loans AS
(SELECT * FROM horizon_bank.loans
WHERE Personal_Loan = "YES")
 , H_loans AS
(SELECT * FROM horizon_bank.loans
WHERE Housing_Loan = "YES")
SELECT * FROM P_loans
UNION
SELECT * FROM H_loans
ORDER BY Customer_ID;
```

Customer_ID	Housing_Loan	Personal_Loan
502654	yes	no
503138	yes	yes
503833	yes	no
504793	yes	yes
505316	yes	yes

Output Report:

The output is a list of customers with their loans, basically making up the loans table.

Query 23: USING

Objective: The aim is to identify customers aged between 18 and 65.

The query below selects customers aged between 18 and 65, from the "customer_accounts" and joins it to the "customers" tables using Customer_ID as a key.

marketing_campaign | marketing_campaign | SQL File 11* | REPORT 3_DA*

Limit to 1000 rows

```
289 ----------------------------------------------------------------
290 #23 USING
291 SELECT CA.*, Age FROM horizon_bank.customer_accounts CA
292 LEFT JOIN horizon_bank.customers C  USING (Customer_ID)
293 WHERE Age BETWEEN 18 AND 65
294 order by Balance desc;
295
296
297
298
299 ----------------------------------------------------------------
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Customer_ID	Balance	Age
641686	26306	54
550331	23663	33
671603	22856	37
695980	22171	29
685618	16957	38

Output Report:

The output is a distribution of customer_IDs and balances together with their ages. Identifying customers with a specific balance threshold is essential for offering tailored financial products and services.

Query 24: WHERE

Objective: The objective is to identify customers aged above 65 years. This is to enable us obtain a view of customers who do not meet the age criteria for qualifying for the new product.

The query below selects customers whose age is greater than 65.

marketing_campaign marketing_campaign SQL File 11* REPORT 3_DA*

Limit to 1000 rows

```
301  ----------------------------------------------------------------------------
302  #24 WHERE
303  SELECT * FROM horizon_bank.customers
304  WHERE Age > 65
305  ORDER BY Age;
306
307
308
309
310
```

Result Grid Filter Rows: Export: Wrap Cell Content:

Customer_ID	MaritalStatus	First_Name	Last_Name	Nationality	Age	Job	Education
637384	Single	Gabrielle	Hodges	US	69	retired	tertiary
752033	Married	Luna	Oliver	US	69	retired	tertiary
758594	Single	Otis	Olsen	US	69	retired	other
608991	Married	Allyson	McDowell	US	70	retired	secondary
623940	Married	Krew	Hull	US	70	retired	tertiary

Output Report:

The output revealed 30 customers who are aged above 65 years, most of whom have retired from active business. The bank can provide specialized services, including retirement financial planning, low-risk investments, healthcare financing options and personalized customer service. These services aim to address the unique financial needs and preferences of elderly customers and ensure their financial well-being during retirement.

Query 25: (Subquery in the FROM clause)

Objective: This aim is to calculate the average duration of customer contacts per customer in minutes.

This query calculates the total duration of customer contacts in minutes by summing the

"Duration_Second" field from the subquery and dividing by 60 to convert if from seconds to minutes and then divides the answer by 127 which is the number of people who were contacted.

```
311     ------------------------------------------------------------
312     #25 Subquery in the FROM clause
313     SELECT
314         (SUM(Duration_Second) / 60)/127 AS Duration_Minute
315     FROM
316         (SELECT
317             *
318         FROM
319             horizon_bank.contacts) AS Cust_contacts;
320
```

Duration_Minute
21.97913386

Output Report:

We can infer that, on average, the bank dedicated approximately 21 minutes to engage with each of the 127 customers it contacted. This suggests that the bank values customer interactions and invests time in providing personalized services or addressing their needs. It may also indicate a commitment to quality customer service and a willingness to spend adequate time to address customer inquiries, concerns, or transactions.

Query 26 (Subquery in the WHERE clause)

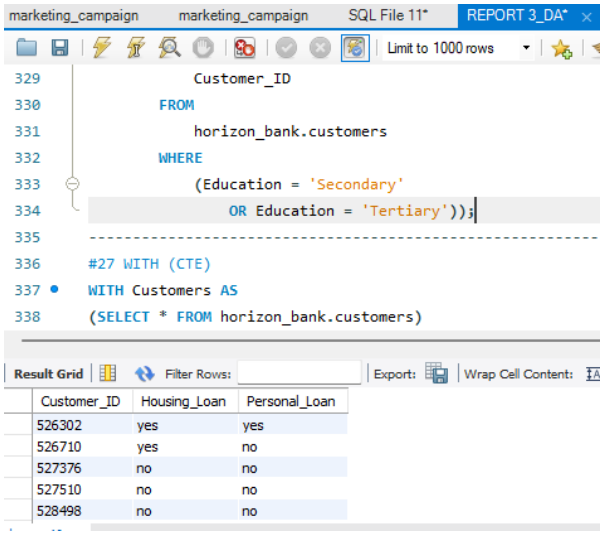
Objective: The aim here is to identify customers with specific educational backgrounds who have loans. Identifying customers with particular educational backgrounds who have loans is valuable for understanding the loan profiles of different education levels.

The query below relies on a subquery within the WHERE clause to filter the data customers with

specific educational backgrounds who have loans.

#26 Subquery in the WHERE clause

```
SELECT
   *
FROM
   horizon_bank.loans AS L
WHERE
   L.Customer_ID IN (SELECT
         Customer_ID
      FROM
         horizon_bank.customers
      WHERE
         (Education = 'Secondary'
            OR Education = 'Tertiary'));
```



Output Report:

Surprisingly, the majority of customers with educational level as Secondary or Tertiary do not have any of the loans in this category. It could be that they may have alternative funding sources, such as scholarships, grants, or family support, which reduces the need for loans.

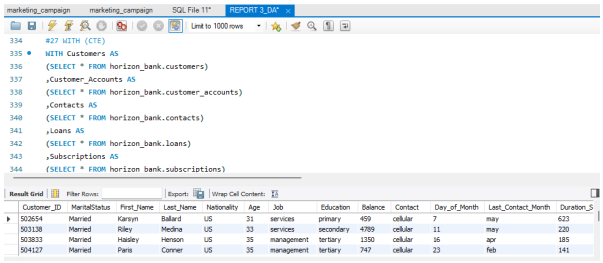
Query 27: WITH - CTE

Objective: The aim is to create a comprehensive dataset by joining multiple tables. This simplifies data analysis and provides a holistic view of customer profiles.

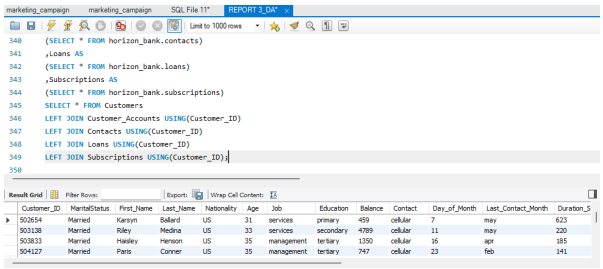
The query below creates a comprehensive dataset by combining customers table, customer accounts table, contact table, loan table, and subscription table using Common Table Expressions (CTE).

#27 WITH (CTE)

```
WITH Customers AS
(SELECT * FROM horizon_bank.customers)

,Customer_Accounts AS
(SELECT * FROM
horizon_bank.customer_accounts)

,Contacts AS
(SELECT * FROM horizon_bank.contacts)

,Loans AS
(SELECT * FROM horizon_bank.loans)

,Subscriptions AS
(SELECT * FROM
horizon_bank.subscriptions)

SELECT * FROM Customers
LEFT JOIN Customer_Accounts
USING(Customer_ID)
LEFT JOIN Contacts USING(Customer_ID)
LEFT JOIN Loans USING(Customer_ID)
LEFT JOIN Subscriptions
USING(Customer_ID);
```



continued…



Output Report:

The output provides a simplified table of customers data for easy querying and data analysis.

Query 28: Window Function

Objective: The objective is to assign row numbers within priority status categories to help in

ranking and identifying customers based on their financial status.

The query below assigns row numbers to customers within priority status categories. It is partitioned by priority status and orders by priority status and balance.

```
WITH P_status AS
(
SELECT Customer_ID, Balance, CASE WHEN Balance > 20000 THEN "High_Priority"
WHEN Balance BETWEEN 5200 AND 20000 THEN "Low_Priority"
ELSE "Disqualified"
END Priority_Status
FROM horizon_bank.customer_accounts
)
SELECT ROW_NUMBER() OVER (PARTITION BY Priority_Status ORDER BY Priority_Status, Balance ) AS RowNumber,
Customer_ID, First_Name, Last_Name, Balance, Priority_Status FROM P_status
LEFT JOIN horizon_bank.Customers USING(Customer_ID);
```

RowNumber	Customer_ID	First_Name	Last_Name	Balance	Priority_Status
1	695980	Reign	Preston	22171	High_Priority
2	671603	Cora	Velez	22856	High_Priority
3	550331	Colten	Dean	23663	High_Priority
4	641686	Novah	Pierce	26306	High_Priority
1	566745	Matilda	Tran	5291	Low_Priority

Output Report:

The ranking from the table above will help the bank in ranking and identifying customers based on their financial status.

What Next?

Marketing Campaign

Customers who qualify for the product and subscribed to marketing campaigns (and are marked as "Can_Call") should be contacted by phone while those who requested not to be called should be contacted via email for targeted marketing.

Reflection

Reflecting on the bank's current policy that requires a customer to have existing loans to qualify for the new term deposit product seems to exclude a significant portion of potential customers. This approach may be biased against those without loans but with deposit needs. In response, the bank should consider revising this requirement or offering alternatives to cater to these customers' financial needs. For those without loans but with deposit requirements, the bank could provide attractive deposit rates, making it more inclusive.

We also realized that most of the customers of Horizon bank have low balances which currently disqualifies them from accessing the new product. To address this, the bank should engage with these customers to better understand their specific needs. It can create tailored products or establish value chains that redirect customer spending, ensuring more funds remain within the bank's ecosystem. By identifying the spending habits and preferences of its customers, the bank can offer more targeted solutions.

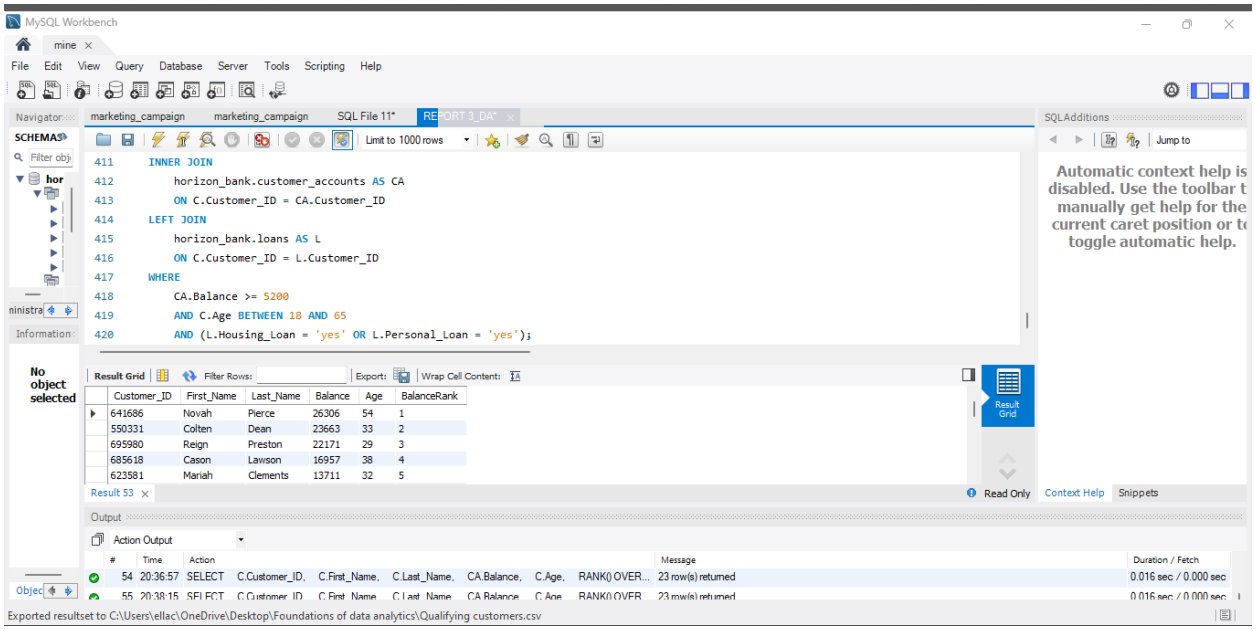
Additionally, offering various subscription options can help expand its contact customer base.

Lastly, given the fact that we had to follow the order of the 28 queries, we were restricted with the output we could generate from a query at different points in time. This had an impact on the consistency of the story we sought to tell with the output of our query given our objective of creating a list of customers who meet the requirements set out by Horizon bank in order to qualify for its new term deposit product.

Nonetheless, below is the query and screenshot of the output of the 23 customers who meet all 3 requirements set by the bank in order to qualify for the new term deposit product ranked in order of priority using their account balances.

```
SELECT
  C.Customer_ID,
  C.First_Name,
  C.Last_Name,
  CA.Balance,
```

```
    C.Age,
    RANK() OVER (ORDER BY CA.Balance DESC) AS BalanceRank
FROM
    horizon_bank.customers AS C
INNER JOIN
    horizon_bank.customer_accounts AS CA
    ON C.Customer_ID = CA.Customer_ID
LEFT JOIN
    horizon_bank.loans AS L
    ON C.Customer_ID = L.Customer_ID
WHERE
    CA.Balance >= 5200
    AND C.Age BETWEEN 18 AND 65
    AND (L.Housing_Loan = 'yes' OR L.Personal_Loan = 'yes');
```


MySQL Workbench
mine
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
marketing_campaign
marketing_campaign
SQL File 11*
REPORT 3_DA*
Limit to 1000 rows
INNER JOIN
horizon_bank.customer_accounts AS CA
ON C.Customer_ID = CA.Customer_ID
LEFT JOIN
horizon_bank.loans AS L
ON C.Customer_ID = L.Customer_ID
WHERE
CA.Balance >= 5200
AND C.Age BETWEEN 18 AND 65
AND (L.Housing_Loan = 'yes' OR L.Personal_Loan = 'yes');
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
No object selected
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Customer_ID First_Name Last_Name Balance Age BalanceRank
641686 Novah Pierce 26306 54 1
550331 Colten Dean 23663 33 2
695980 Reign Preston 22171 29 3
685618 Cason Lawson 16957 38 4
623581 Mariah Clements 13711 32 5
Result 53
Read Only
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
54 20:36:57 SELECT C.Customer_ID, C.First_Name, C.Last_Name, CA.Balance, C.Age, RANK() OVER... 23 row(s) returned 0.016 sec / 0.000 sec
Exported resultset to C:\Users\ellac\OneDrive\Desktop\Foundations of data analytics\Qualifying customers.csv