Objective Answers:

1. **What is the distribution of account balances across different regions?**

Query:

```
select
       t3.GeographyLocation,
       sum(Balance) as TotalBalance,
    avg(Balance) as AvgBalanece
from bank_churn t1
join customerinfo t2
on t1.CustomerID=t2.CustomerID
join geography t3
on t2.GeographyID=t3.GeographyID
group by t3.GeographyLocation;
```

Output:

GeographyLocation	TotalBalance	AvgBalanece
Spain	153123552.01	61818.147763
France	311332479.49	62092.636516
Germany	300402861.38	119730.116134

Brief Answer Explanation:

The analysis of account balance distribution across regions reveals key differences in customer financial profiles.

- **France** records the **highest total balance** (approximately **€311.33 million**), indicating a large customer base or higher cumulative deposits.
- **Germany** follows with a total of around **€300.40 million**, while **Spain** shows a comparatively lower total balance of **€153.12 million**.
- However, when examining **average balances per customer**, **Germany** leads with an average of approximately **€119.73K**, suggesting that German customers generally maintain higher individual account balances compared to those in France and Spain.
- This insight highlights regional variations in customer wealth and engagement, which can guide the bank’s targeted marketing and financial strategies.

2. **Identify the top 5 customers with the highest Estimated Salary in the last quarter of the year. (SQL)**

Query:

```
select
       CustomerID,
    EstimatedSalary
from customerinfo
activecustomer
where month(Bank_DOJ) in (10,11,12)
order by EstimatedSalary desc
limit 5;
```

Output:

CustomerID	EstimatedSalary
15634359	199970.74
15804211	199841.32
15687913	199805.63
15763065	199753.97
15599792	199638.56
NULL	NULL

Brief Answer Explanation

- The objective of this analysis is to identify the top 5 high-value customers based on their Estimated Salary during the last quarter (October–December) of the year.
- The query filters the `customerinfo` dataset to include only customers who joined the bank in months 10, 11, and 12 using the `month(Bank_DOJ)` function.
- Results are sorted in descending order of Estimated Salary to rank the highest earners first.
- The `LIMIT 5` clause ensures only the top 5 customers are returned.
- From the result, these customers have Estimated Salaries close to $200,000, indicating they are likely high-value clients for targeted marketing or personalized banking services.
- This insight can help the bank in strategic decision-making, such as offering premium services, investment opportunities, or loyalty programs to retain and engage these valuable customers.

3. Calculate the average number of products used by customers who have a credit card. (SQL)

Query:

```
select Round(avg(NumOfProducts),2) as
AvgNumOfProducts
from bank_churn
where CreditID = 1;
```

Output:

AvgNumOfProducts
1.53

Brief Answer Explanation

- The query calculates the average number of products held by customers who possess a credit card (`CreditID = 1`).
- The `AVG()` function is used to determine the mean number of products per customer in this specific group.
- The `ROUND()` function rounds the result to two decimal places for clarity.
- The resulting value of 1.53 indicates that, on average, customers with credit cards use approximately 1 to 2 banking products.
- This insight helps the bank understand product engagement levels among credit card holders and identify opportunities to cross-sell additional products to increase customer value and retention.

4. Calculate the average number of products used by customers who have a credit card. (SQL)

Query:

```
with most_recent_year as(
select max(year(Bank_DOJ)) as  latest_year from customerinfo),

churn_data as(select
    GenderCategory,
    sum(case when t1.ExitID = 1 then 1 else 0 end) as
ExitCustomerCount,
    count(*) as TotalCustomerCount
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
join exitcustomer t3
on t1.ExitID = t3.ExitID
join gender t4
on t2.GenderID = t4.GenderID
where year(Bank_DOJ) = (select latest_year from
most_recent_year)
group by GenderCategory)
select
       GenderCategory,
       ExitCustomerCount,
    TotalCustomerCount,
    round((100*ExitCustomerCount/TotalCustomerCount),2) as
Churn_Rate
from churn_data
group by GenderCategory;
```

Output:

GenderCategory	ExitCustomerCount	TotalCustomerCount	Churn_Rate
Female	385	1537	25.05
Male	273	1776	15.37

Brief Answer Explanation

- This query analyzes customer churn rates by gender for the most recent year of customer data.
- A Common Table Expression (CTE) named `most_recent_year` identifies the latest year based on the customers' joining date (Bank_DOJ).
- Another CTE, `churn_data`, aggregates the number of customers who exited (`ExitID` = 1) and the total customers by gender for that year.
- The final query calculates the churn rate as a percentage using the formula: 100 * (`ExitCustomerCount` / `TotalCustomerCount`).
- Results show that female customers have a higher churn rate (25.05%) compared to male customers (15.37%).
- This insight indicates that female customers are more likely to leave the bank, highlighting a potential area where the bank can investigate reasons for dissatisfaction and develop targeted retention strategies to improve loyalty.

5. Which Compare the average credit score of customers who have exited and those who remain.

Query:

```
select t1.ExitID,ExitCategory,
       avg(CreditScore)  as AvgCreditScore
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
join exitcustomer t3
on t1.ExitID = t3.ExitID
group by ExitID,ExitCategory;
```

Output:

ExitID	ExitCategory	AvgCreditScore
0	Retain	651.8532
1	Exit	645.3515

Brief Answer Explanation

- The query compares the average credit scores of customers who have exited (`ExitID = 1`) versus those who remain with the bank (`ExitID = 0`).
- It uses the `AVG()` function to calculate the mean credit score for each group, joining the bank_churn, `customerinfo, and exitcustomer` tables to access both churn status and credit data.
- The results show that retained customers have a slightly higher average credit score (651.85) compared to exited customers (645.35).
- This indicates that customers with lower credit scores are somewhat more likely to leave, suggesting a potential link between financial stability and customer retention.
- The bank can use this insight to design targeted engagement or credit-support programs to help at-risk customers and reduce churn.

6. Which gender has a higher average estimated salary, and how does it relate to the number of active accounts?

Query:

```
select
       t1.GenderID,
       GenderCategory,
    round(avg(EstimatedSalary),2) as AvgEstimatedSalary,
    count(case when t4.ActiveID = 1 then 1 end) as
ActiveAccountsCount
from customerinfo t1
join gender t2
on t1.GenderID = t2.GenderID
join bank_churn t3
on t1.CustomerID=t3.CustomerID
join activecustomer t4
on t3.ActiveID = t4.ActiveID
group by t1.GenderID,GenderCategory;
```

Output:

GenderID	GenderCategory	AvgEstimatedSalary	ActiveAccountsCount
2	Female	100601.54	2284
1	Male	99664.58	2867

Brief Answer Explanation

- This query evaluates the average estimated salary by gender and examines its relationship with the number of active customer accounts.
- It joins the `customerinfo`, `gender`, `bank_churn`, and `activecustomer` tables to combine demographic, financial, and activity-related data.
- The AVG(`EstimatedSalary`) function calculates the mean income for each gender, while the `COUNT()` with a condition on `ActiveID = 1` counts how many customers are actively using their accounts.
- Results show that female customers have a slightly higher average estimated salary (₹100,601.54) compared to male customers (₹99,664.58).
- However, male customers have more active accounts (2,867) than females (2,284).
- This suggests that income level alone doesn't determine customer engagement — other factors like product preferences, service satisfaction, or banking habits may influence account activity levels.

7. Segment the customers based on their credit score and identify the segment with the highest exit rate.

Query:

```
select
       Segment,
       sum(case when ExitID = 1 then 1 else 0 end) as exitcustcount,
    count(*) as total_cust_count,
    round((100*sum(case when ExitID = 1 then 1 else 0
end)/count(*)),2) as exit_rate
from(
select
       CustomerID,
    ExitID,
    case
              when CreditScore between 800 and 850 then "Exceptional"
        when CreditScore between 740 and 799 then "Very Good"
        when CreditScore between 670 and 739 then "Good"
        when CreditScore between 580 and 669 then "Fair"
        when CreditScore between 300 and 579 then "Poor"
    end as Segment
from bank_churn)p1
group by Segment
order by exit_rate desc;
```

Output:

Segment	exitcustcount	total_cust_count	exit_rate
Poor	520	2362	22.02
Very Good	252	1224	20.59
Fair	685	3331	20.56
Exceptional	128	655	19.54
Good	452	2428	18.62

Brief Answer Explanation

- This query segments customers into credit score categories—Poor, Fair, Good, Very Good, and Exceptional—and calculates the exit (churn) rate for each segment.
- A CASE statement classifies each customer based on their credit score range, and the query then counts total customers and exited customers (ExitID = 1) per segment.
- The exit rate is computed as the percentage of exited customers out of the total in each group.
- Results reveal that the "Poor" credit score segment has the highest churn rate (22.02%), followed by the "Very Good" (20.59%) and "Fair" (20.56%) segments.
- This suggests that customers with low credit scores are most likely to leave, possibly due to financial stress or limited access to credit products.
- Interestingly, the "Very Good" segment's relatively high churn rate indicates that even financially stable customers may exit, highlighting the need for improved retention strategies across all credit tiers, not just low scorers.

8. **Find out which geographic region has the highest number of active customers with a tenure greater than 5 years.**

Query:

```
select
    t1.GeographyID,
    t3.GeographyLocation,
    count(t1.CustomerID) as ActiveCustomerCount
from customerinfo t1
join bank_churn t2
on t1.CustomerID= t2.CustomerID
join geography t3
on t1.GeographyID=t3.GeographyID
where ActiveID = 1 and t2.Tenure > 5
group by t1.GeographyID,GeographyLocation
order by ActiveCustomerCount desc;
```

Output:

GeographyID	GeographyLocation	ActiveCustomerCount
1	France	797
2	Spain	431
3	Germany	399

Brief Answer Explanation

- This query identifies which geographic region has the highest number of active customers who have been with the bank for more than 5 years.
- It joins the `customerinfo`, bank_churn, and geography tables to combine customer activity, tenure, and location data.
- The WHERE clause filters for active customers (`ActiveID` = 1) with tenure greater than 5 years, ensuring that only long-term active clients are considered.
- The COUNT() function then aggregates the number of qualifying customers per region, and the results are sorted in descending order of active customer count.
- The results show that France leads with 797 active long-tenure customers, followed by Spain (431) and Germany (399).
- This indicates that France has the most stable and loyal customer base, suggesting stronger customer relationships or more effective retention strategies in that region.

9. What is the impact of having a credit card on customer churn, based on the available data?

Query:

```
select
       t2.Category,
    count(*) as TotalCustomer,
    sum(case when ExitID = 1 then 1 else 0 end) as ExitCustomer,
    round((100*sum(case when ExitID = 1 then 1 else 0
end)/count(*)),2) as chrun_rate
from bank_churn t1
join creditcard t2
on t1.CreditID = t2.CreditID
group by t2.Category
order by chrun_rate desc;
```

Output:

Category	TotalCustomer	ExitCustomer	chrun_rate
non credit card holder	2945	613	20.81
credit card holder	7055	1424	20.18

Brief Answer Explanation

- This query evaluates the relationship between credit card ownership and customer churn.
- It joins the bank_churn and `creditcard` tables to categorize customers as credit card holders or non-holders, then calculates the total customers, number of exited customers, and churn rate for each group.
- The churn rate is derived by dividing the number of exited customers by the total customers in each category and multiplying by 100 for a percentage.
- Results indicate that non–credit card holders have a slightly higher churn rate (20.81%) compared to credit card holders (20.18%).
- This suggests that having a credit card is marginally associated with better customer retention, possibly because credit card users are more engaged with the bank's services or benefit from loyalty programs and convenience features that encourage continued banking relationships.

10. For customers who have exited, what is the most common number of products they have used?

Query:

```
select
       NumOfProducts,
    count(*) as CommonProductsCustomerCount
from bank_churn
where ExitID = 1
group by NumOfProducts
order by CommonProductsCustomerCount desc;
```

Output:

NumOfProducts	CommonProductsCustomerCount
1	1409
2	348
3	220
4	60

Brief Answer Explanation

- This query identifies the most common number of products used by customers who have exited the bank (`ExitID = 1`).
- It groups the exited customers by `NumOfProducts` and counts the number of customers in each group using `COUNT()`.
- The results show that customers who used only 1 product are the most likely to exit (1,409 customers), followed by 2, 3, and 4 products.
- This indicates that limited product engagement is associated with higher churn, suggesting that customers with more diverse product usage are more loyal.
- The insight highlights an opportunity for the bank to promote cross-selling and encourage customers to adopt additional products to reduce churn risk.

11. Examine the trend of customers joining over time and identify any seasonal patterns (yearly or monthly). Prepare the data through SQL and then visualize it.

Query:

```
select
        year(Bank_DOJ) as Year,
    count(CustomerID) as
YearlyJoinedCust
from customerinfo
group by year(Bank_DOJ)
order by Year;
```

```
select
        month(Bank_DOJ) as Month,
    monthname(Bank_DOJ) as MonthName,
    count(CustomerID) as MonthlyJoinedCust
from customerinfo
group by
month(Bank_DOJ),monthname(Bank_DOJ)
order by Month;
```

Output:

Year	YearlyJoinedCust
2016	1951
2017	2143
2018	2593
2019	3313

Month	MonthName	MonthlyJoinedCust
1	January	482
2	February	324
3	March	654
4	April	653
5	May	694
6	June	765
7	July	721
8	August	674
9	September	1356
10	October	831
11	November	1437
12	December	1409

Brief Answer Explanation

- These queries analyze the trend of customer acquisitions over time to identify yearly and monthly patterns.

Yearly Trend:

- The first query groups customers by year of joining (Bank_DOJ) and counts the number of customers who joined each year.
- Results show a steady increase in new customers from 2016 (1,951) to 2019 (3,313), indicating growing customer acquisition over time, which may reflect successful marketing strategies or expansion efforts.

Monthly Trend:

- The second query groups customers by month of joining to explore seasonal patterns.
- Results reveal peaks in September (1,356), November (1,437), and December (1,409), suggesting that end-of-quarter and holiday periods attract more new customers, possibly due to promotions, salary cycles, or festive campaigns.
- These insights can help the bank plan marketing campaigns and resource allocation effectively, targeting months with historically higher customer sign-ups to maximize acquisition efficiency.

12. Analyze the relationship between the number of products and the account balance for customers who have exited.

Query:

```
select
       NumOfProducts,
    count(*) as CommonProductsCustomerCount,
     Avg(Balance) as AvgAccountBalance,
      sum(Balance) as TotalAccountBalance
from bank_churn t1
where ExitID = 1
group by NumOfProducts
order by AvgAccountBalance desc;
```

Output:

NumOfProducts	CommonProductsCustomerCount	AvgAccountBalance	TotalAccountBalance
4	60	93733.135000	5623988.10
1	1409	92028.819787	129668607.08
2	348	90252.357155	31407820.29
3	220	85853.087091	18887679.16

Brief Answer Explanation

- This query examines the relationship between the number of products held and the account balance for customers who have exited the bank (`ExitID = 1`).
- It groups exited customers by `NumOfProducts` and calculates:
  - Count of customers per product category (`CommonProductsCustomerCount`)
  - Average account balance (`AvgAccountBalance`)
  - Total account balance (`TotalAccountBalance`)
- Results show that:
  - Customers with 4 products have the highest average balance (₹93,733) but represent a small group (60 customers).
  - Customers with 1 product form the largest exited group (1,409 customers) with a slightly lower average balance (₹92,029).
  - Those with 2 or 3 products have lower average balances.
- Insight: While higher product usage correlates with higher individual balances, the largest churn occurs among single-product customers, suggesting that limited engagement, rather than account balance, is a stronger driver of churn.
- The bank can use this to encourage multi-product adoption to improve retention among at-risk customers.

13. Identify any potential outliers in terms of balance among customers who have remained with the bank

Query:

```
select
       NumOfProducts,
    count(*) as CommonProductsCustomerCount,
     Avg(Balance) as AvgAccountBalance,
      sum(Balance) as TotalAccountBalance
from bank_churn t1
where ExitID = 1
group by NumOfProducts
order by AvgAccountBalance desc;
```

Output:

No Output.

Brief Answer Explanation

- This query identifies potential outliers in account balances among active customers (`ActiveID` = 1) using the Interquartile Range (IQR) method.
- Steps involved:
  - Final CTE: Assigns a row number to each active customer based on their balance and counts total customers.
  - `First_Quartile` & `Third_Quartile` CTEs: Determine the 25th percentile (Q1) and 75th percentile (Q3) balances.
  - `InterQuartileRange` CTE: Calculates the IQR = Q3 - Q1.
  - `Lower_Limit & Upper_Limit` CTEs: Compute thresholds for detecting outliers:
    - Lower Limit = Q1 - 1.5 × IQR
    - Upper Limit = Q3 + 1.5 × IQR
  - Final selection checks which balances fall outside these limits.
- Result shows no outliers, indicating that the account balances of active customers are fairly consistent and there are no extreme anomalies.
- This suggests a stable financial distribution among retained customers, which can help the bank make reliable decisions regarding account management and risk assessment.

14. How many different tables are given in the dataset, out of these tables which table only consists of categorical variables?

15. Using SQL, write a query to find out the gender-wise average income of males and females in each geography id. Also, rank the gender according to the average value.

Query:

```
select
       t1.GenderID,
    t2.GenderCategory,
    t1.GeographyID,
    t3.GeographyLocation,
    t1.AvgIncome,
    rank() over(partition by GenderCategory order by AvgIncome desc)
as rnk
from(
select
       GenderID,
    GeographyID,
    Round(avg(EstimatedSalary),2) as AvgIncome
from customerinfo
group by GenderID,GeographyID) t1
join Gender t2
on t1.GenderID = t2.GenderID
join Geography t3
on t1.GeographyID= t3.GeographyID
order by AvgIncome desc;
```

Output:

GenderID	GenderCategory	GeographyID	GeographyLocation	AvgIncome	rnk
2	Female	3	Germany	102446.42	1
2	Female	2	Spain	100734.11	2
1	Male	1	France	100174.25	1
1	Male	3	Germany	99905.03	2
2	Female	1	France	99564.25	3
1	Male	2	Spain	98425.69	3

Brief Answer Explanation

- This query calculates the average estimated income for male and female customers across each geographic region and ranks them within their gender category.
- Steps:
  - The subquery (t1) groups customers by GenderID and GeographyID and calculates the average estimated salary.
  - It joins with the Gender and Geography tables to fetch gender names and geographic locations.
  - The RANK() function partitions by GenderCategory and ranks geographies based on average income in descending order.
- Results indicate:
  - Among females, Germany has the highest average income (₹102,446.42), followed by Spain and France.
  - Among males, France ranks highest (₹100,174.25), followed by Germany and Spain.
- This analysis highlights income variations by gender across regions, helping the bank tailor marketing, product offerings, and financial planning based on regional income demographics.

16. Using SQL, write a query to find out the average tenure of the people who have exited in each age bracket (18-30, 30-50, 50+).

Query:

```
select
       age_bracket,
    Avg(Tenure) as AvgTenure
from(
select
       t1.CustomerID,
       Age,
    Tenure,
    (case
              when Age between 18 and 30 then "18-30"
        when Age between 31 and 50 then "31-50"
        when Age >50 then "50+"
    end) as age_bracket
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
where ExitID = 1)p
group by age_bracket
order by age_bracket;
```

Output:

age_bracket	AvgTenure
18-30	4.7770
31-50	4.8899
50+	4.8330

Brief Answer Explanation

- This query calculates the average tenure of customers who have exited the bank (ExitID = 1) across different age brackets.
- Steps:
  - A CASE statement categorizes customers into age groups: 18–30, 31–50, and 50+.
  - The subquery joins bank_churn and customerinfo to get both age and tenure information.
  - The outer query groups the data by age bracket and calculates the average tenure using AVG().
- Results indicate that:
  - Customers aged 31–50 have the highest average tenure before exiting (4.89 years).
  - Customers in 18–30 and 50+ age groups have slightly lower average tenures (4.78 and 4.83 years, respectively).
- This suggests that middle-aged customers tend to stay slightly longer before leaving, which can help the bank design targeted retention strategies for different age groups.

17. Is there any direct correlation between salary and the balance of the customers? And is it different for people who have exited or not?

⋙ Query:

```sql
SELECT
       t1.CustomerID,
    t2.Surname as CustomerName,
    EstimatedSalary AS
CustomerSalary,
    Balance AS CustomerBalance,
    t3.ExitCategory
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
join exitcustomer t3
on t1.ExitID = t3.ExitID
where t1.ExitID = 1
order by CustomerBalance desc;
```

```sql
SELECT
       t1.CustomerID,
    t2.Surname as CustomerName,
    EstimatedSalary AS
CustomerSalary,
    Balance AS CustomerBalance,
       t3.ExitCategory
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
join exitcustomer t3
on t1.ExitID = t3.ExitID
where t1.ExitID = 0
order by CustomerBalance desc;
```

Output:

Co	Customer	Customer	Customer	ExitCategory		
15757408	Lo	81054	250898.1	Exit		
15715622	To Rot	147965	238387.6	Exit		
15714241	Haddon	101108.9	222267.6	Exit		
15586674	Shaw	74176.71	216109.9	Exit		
15594408	Chia	75161.25	213146.2	Exit		
15671256	Macartney	188574.1	211774.3	Exit		
15736420	Macdonal	197297.8	210433.1	Exit		
15721658	Fleming	150694.4	209767.3	Exit	**Correlation =**	**0.022557**
15578671	Webb	133267.7	209490.2	Exit		
15709920	Burke	50774.81	208165.5	Exit		
15769412	Atkinson	157694.8	207035	Exit		
15784180	Ku	46632.87	206329.7	Exit		
15607275	Ch'ang	42774.84	206014.9	Exit		
15673020	Smith	738.88	204510.9	Exit		
15588670	Despeissi	179978.7	203715.2	Exit		
15620756	Stokes	17298.72	202904.6	Exit		
15689514	Kang	133020.9	201696.1	Exit		
15793688	Bancks	158032.5	201009.6	Exit		
15776545	Napolitan	82872.64	200725	Exit		
15648489	Ting	24207.86	199689.5	Exit		
15725665	Lo	191198.9	198546.1	Exit		

Customer	Customer	Customer	Customer	ExitCategory		
15571958	McIntosh	171867.1	221532.8	Retain		
15599131	Dilke	128815.3	214347	Retain		
15769818	Moore	69372.88	212778.2	Retain		
15620268	Thomson	115268.9	212696.3	Retain		
15780212	Mao	176395	212693	Retain	**Correlation =**	**0.030165**
15690589	Udinesi	148814.5	212314	Retain		
15795298	Olisaemel	102986.2	206868.8	Retain		
15627971	Coates	16281.94	206663.8	Retain		
15664498	Golovanov	156424.4	205962	Retain		
15745433	Conti	65464.66	205770.8	Retain		
15746664	Ts'ui	128268.4	204223	Retain		
15663888	Connor	109538.4	204017.4	Retain		
15620570	Sinnett	72375.03	202443.5	Retain		
15668818	Chiduben	113244.7	200322.5	Retain		
15807245	McKay	94142.35	200117.8	Retain		
15737509	Morrison	68106.29	199229.1	Retain		
15594270	Biryukov	14278.18	198338.8	Retain		
15766458	Tang	69664.35	198113.9	Retain		
15666559	Gould	116124.3	197715.9	Retain		
15664506	Goodwin	52710.7	197436.8	Retain		
15680683	Simmons	141453.6	197200	Retain		

Brief Answer Explanation

- This analysis examines the relationship between customer salary (EstimatedSalary) and account balance for both exited and retained customers.
- Two separate datasets were prepared using SQL:
  - Exited customers (ExitID = 1) – their salaries and balances.
  - Retained customers (ExitID = 0) – their salaries and balances.
- Using Excel's CORREL function, the correlation coefficients were calculated:
  - Exited customers: 0.0226
  - Retained customers: 0.0302
- Both values are very close to zero, indicating almost no linear correlation between salary and account balance in either group.
- Interpretation: Higher salary does not necessarily translate to higher account balances, and this pattern remains consistent regardless of whether customers exit or stay.
- This suggests that other factors—like product usage, spending habits, or financial behavior—likely play a larger role in determining balances than salary alone.

18. Is there any correlation between the salary and the Credit score of customers?

Query:

```
SELECT
       t1.CustomerID,
    t2.Surname as CustomerName,
    EstimatedSalary AS CustomerSalary,
    CreditScore,
       Category
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
join creditcard t3
on t1.CreditID = t3.CreditID
order by CustomerSalary desc;
```

Output:

CustomerID	CustomerName	CustomerSalary	CreditScore	Category			
15662021	Lucciano	199992.48	685	non credit card holder			
15634359	Dyer	199970.74	639	credit card holder			
15697270	Gannon	199953.33	608	credit card holder			
15762331	Moss	199929.17	569	credit card holder			
15709136	Adams	199909.32	620	credit card holder			
15741719	DeRose	199862.75	540	non credit card holder			
15682834	Johnstone	199857.47	715	credit card holder			
15804211	Oluchukwu	199841.32	719	credit card holder			
15815656	Hopkins	199808.1	541	credit card holder		**Correlation =**	**-0.02302**
15687913	Mai	199805.63	501	non credit card holder			
15720713	Chibueze	199775.67	850	credit card holder			
15772601	Lu	199761.29	845	non credit card holder			
15763065	Palerma	199753.97	700	credit card holder			
15712772	Onwubiko	199727.72	757	credit card holder			
15661670	Chidozie	199725.39	524	credit card holder			
15672152	Grant	199693.84	850	non credit card holder			
15782758	Ozerova	199674.83	632	credit card holder			
15698474	Sagese	199661.5	601	credit card holder			
15814040	Munroe	199657.46	610	credit card holder			
15743040	Kuznetsova	199645.45	724	non credit card holder			
15642885	Gray	199644.2	792	credit card holder			

Brief Answer Explanation

- This analysis investigates the correlation between customer salary (EstimatedSalary) and credit score.
- The query retrieves salary and credit score information for all customers, including whether they hold a credit card.
- Using Excel's CORREL function, the correlation coefficient was calculated as -**0.0230**, which is very close to zero.
- Interpretation: There is virtually no linear relationship between salary and credit score among customers.
- This indicates that a higher income does not necessarily correspond to a higher credit score, suggesting that other factors like repayment behavior, credit history, or outstanding debts play a more significant role in determining creditworthiness than salary alone.

19. Rank each bucket of credit score as per the number of customers who have churned the bank.

Query:

```
select *,
              rank() over(order by ChurnedCustomerCount) as ChurnedRank
from(
select
       Segment,
    count(*) as ChurnedCustomerCount
from(
select
       CustomerID,
    case
              when CreditScore between 800 and 850 then "Exceptional"
        when CreditScore between 740 and 799 then "Very Good"
        when CreditScore between 670 and 739 then "Good"
        when CreditScore between 580 and 669 then "Fair"
        when CreditScore between 300 and 579 then "Poor"
```

Output:

Segment	ChurnedCustomerCount	ChurnedRank
Exceptional	128	1
Very Good	252	2
Good	452	3
Poor	520	4
Fair	685	5

Brief Answer Explanation

- This query ranks credit score segments based on the number of customers who have exited (churned) the bank.
- Steps:
  - A CASE statement classifies customers into credit score segments: Exceptional, Very Good, Good, Fair, and Poor.
  - The subquery counts the number of churned customers (ExitID = 1) in each segment.
  - The outer query uses RANK() to assign a rank based on churned customer count, with the lowest count getting rank 1.
- Results show:
  - Exceptional (128 churned customers) – Rank 1
  - Very Good (252) – Rank 2
  - Good (452) – Rank 3
  - Poor (520) – Rank 4
  - Fair (685) – Rank 5
- Insight: Customers in the “Fair” segment have the highest churn, while “Exceptional” customers churn the least.
- This helps the bank identify at-risk credit score groups and develop targeted retention strategies for segments with higher churn.

20. According to the age buckets find the number of customers who have a credit card. Also retrieve those buckets that have lesser than average number of credit cards per bucket

Query:

```
with final as(select
        age_bracket,
    count(CustomerID) as cust_count
from(
select
        t1.CustomerID,
        Age,
    Tenure,
    (case
               when Age between 18 and 30 then "18-30"
        when Age between 31 and 50 then "31-50"
        when Age >50 then "50+"
    end) as age_bracket
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
where t1.CreditID = 1)l
group by age_bracket)
select age_bracket, cust_count
from final
where cust_count<(select avg(cust_count) from final);
```

Output:

age_bracket	cust_count
18-30	1400
50+	874

Brief Answer Explanation

- This query analyzes the distribution of credit card holders across different age buckets and identifies buckets with below-average credit card adoption.
- Steps:
- A CASE statement categorizes customers into age groups: 18–30, 31–50, and 50+.
- The subquery filters only customers who hold a credit card (CreditID = 1) and counts the number of credit card holders per age bracket.
- The outer query compares each bucket's count against the average number of credit card holders and selects buckets below this average.
- Results show that the 18–30 (1,400 holders) and 50+ (874 holders) age groups have fewer credit card holders than the average, while the 31–50 group exceeds the average.
- Insight: Younger and older customers are less likely to hold a credit card, suggesting that the bank could design targeted campaigns or incentives to increase credit card adoption in these age groups.

21. Rank the Locations as per the number of people who have churned the bank and average balance of the customers.

Query:

```
with final as(select
       t2.GeographyID,
       t3.GeographyLocation,
    count(*) as NoOfCustomer,
    avg(t1.Balance) as AvgBalance
from bank_churn t1
join customerinfo t2
on t1.CustomerID = t2.CustomerID
join Geography t3
on t2.GeographyID = t3.GeographyID
where ExitID = 1
group by t2.GeographyID,t3.GeographyLocation)

select
       GeographyID,
    GeographyLocation,
    NoOfCustomer,
    dense_rank() over(order by NoOfCustomer desc) as cust_rank,
    AvgBalance,
    dense_rank() over(order by AvgBalance desc) as balance_rank
from final;
```

Output:

GeographyID	GeographyLocation	NoOfCustomer	cust_rank	AvgBalance	balance_rank
3	Germany	814	1	120361.075590	1
2	Spain	413	3	72513.352446	2
1	France	810	2	71192.795728	3

Brief Answer Explanation

- This query ranks geographic locations based on two criteria for customers who have churned (ExitID = 1):
- Number of churned customers
- Average account balance
- Steps:
  - The final CTE aggregates churned customers by GeographyID and GeographyLocation, calculating the total number of customers and their average balance.
  - The outer query applies DENSE_RANK() to:
  - Rank locations by number of churned customers (cust_rank) in descending order.
  - Rank locations by average balance (balance_rank) in descending order.
- Results:
  - Germany has the highest churn count (814) and the highest average balance (₹120,361).
  - France has slightly fewer churned customers (810) but the lowest average balance (₹71,193).

    - Spain has the lowest churn count (413) but a moderate average balance (₹72,513).
- Insight: High-balance customers in Germany are leaving the bank more frequently, highlighting a need for retention strategies targeting premium clients in that region. The ranking provides a clear priority list for intervention based on both volume and financial impact.

22. As we can see that the "CustomerInfo" table has the CustomerID and Surname, now if we have to join it with a table where the primary key is also a combination of CustomerID and Surname, come up with a column where the format is "CustomerID_Surname"

» **Query:**

```
SELECT
    CustomerID,
    Surname,
    CONCAT(CustomerID, '_', Surname) AS CustomerKey
FROM CustomerInfo
limit 10;
```

Output:

CustomerID	Surname	CustomerKey
15565701	Ferri	15565701_Ferri
15565706	Akobundu	15565706_Akobundu
15565714	Cattaneo	15565714_Cattaneo
15565779	Kent	15565779_Kent
15565796	Docherty	15565796_Docherty
15565806	Toosey	15565806_Toosey
15565878	Bates	15565878_Bates
15565879	Riley	15565879_Riley
15565891	Dipietro	15565891_Dipietro
15565996	Arnold	15565996_Arnold

Brief Answer Explanation

- This query creates a composite key by combining CustomerID and Surname into a single column named CustomerKey.
- The CONCAT() function is used to merge the two columns with an underscore (_) separator, resulting in a format like CustomerID_Surname.
- Example output: 15714241_Haddon
- This approach is useful when joining with another table whose primary key is also a combination of CustomerID and Surname, ensuring unique and consistent matching.
- It simplifies joins and avoids ambiguity, especially in cases where CustomerID or Surname alone may not be unique.

23. Without using "Join", can we get the "ExitCategory" from ExitCustomers table to Bank_Churn table?If yes do this using SQL

Query:

```
select
       CustomerID,
    Balance,
    ExitID,
    (select ExitCategory from exitcustomer where ExitID = t1.ExitID) as
ExitCategory
from bank_churn t1
order by Balance desc
limit 10;
```

Output:

CustomerID	Balance	ExitID	ExitCategory
15757408	250898.09	1	Exit
15715622	238387.56	1	Exit
15714241	222267.63	1	Exit
15571958	221532.80	0	Retain
15586674	216109.88	1	Exit
15599131	214346.96	0	Retain
15594408	213146.20	1	Exit
15769818	212778.20	0	Retain
15620268	212696.32	0	Retain
15780212	212692.97	0	Retain

Brief Answer Explanation

- Yes, it is possible to retrieve the ExitCategory from the ExitCustomer table without using a JOIN by using a correlated subquery.
- In this query, the subquery (`SELECT ExitCategory FROM exitcustomer WHERE ExitID = t1.ExitID`) executes for each row in the bank_churn table, fetching the corresponding ExitCategory.
- The result effectively adds the ExitCategory column to bank_churn based on the ExitID value, similar to what a JOIN would do.
- This approach is useful when a direct join is not preferred or when dealing with simple lookups, although it may be less efficient for very large datasets.
- Example output shows customers with their balance, exit status, and corresponding ExitCategory like Exit or Retain.

24. Were there any missing values in the data, using which tool did you replace them and what are the ways to handle them?

Query:

```
SELECT *
FROM customerinfo
WHERE CustomerID IS NULL
   OR Surname IS NULL
   OR Age IS NULL
   OR GenderID IS NULL
   OR EstimatedSalary IS NULL
   OR GeographyID IS NULL
   OR Bank_DOJ IS NULL;

SELECT *
FROM bank_churn
WHERE CustomerID IS NULL
   OR CreditScore IS NULL
   OR Tenure IS NULL
   OR Balance IS NULL
   OR NumOfProducts IS NULL
   OR CreditID IS NULL
   OR ActiveID IS NULL
   OR ExitID IS NULL;
```

Output:

NO Missing Values in Data

Brief Answer Explanation

- To identify missing values in the dataset, Power BI provides data profiling tools in the Power Query Editor, which can highlight nulls or blanks in each column.
- Once detected, missing values can be handled in several ways depending on the data type and business context:
- Imputation: Replace missing numeric values with mean, median, or mode, and categorical values with most frequent category.
- Forward/Backward Fill: Carry forward or backward the last known value, commonly used in time-series data.
- Default or Placeholder Values: Replace missing entries with a default value such as 0, "Unknown", or "Not Available".
- Removal: Delete rows or columns with missing data, especially if the missing proportion is high and cannot be reliably imputed.
- Using Power BI, you can replace nulls directly in Power Query using the "Replace Values" or "Fill Down/Up" options, making the dataset clean and ready for analysis.

25. Write the query to get the customer IDs, their last name, and whether they are active or not for the customers whose surname ends with "on".

Query:

```
select
       t1.CustomerID,
    Surname,
    ActiveCategory
from customerinfo t1
join bank_churn t2
on t1.CustomerID = t2.CustomerID
join activecustomer t3
on t2.ActiveID=t3.ActiveID
where t2.ActiveID in (0,1)
       and Surname like "%on"
    limit 10;
```

Output:

CustomerID	Surname	ActiveCategory
15566495	Hanson	Inactive Member
15567839	Gordon	Active Member
15568088	Jamieson	Inactive Member
15568360	Rolon	Active Member
15570289	Benson	Inactive Member
15571193	Morrison	Active Member
15571305	Stephenson	Inactive Member
15572145	Ashton	Inactive Member
15572415	Preston	Active Member
15573599	Adamson	Active Member

Brief Answer Explanation

- This query retrieves customer IDs, surnames, and activity status for customers whose surname ends with "on".
- Steps:
  - The customerinfo table provides CustomerID and Surname.
  - The bank_churn table is joined to get the ActiveID, which indicates whether a customer is active.
  - The activecustomer table is joined to convert ActiveID into a readable category like "Active Member" or "Inactive Member."
  - The WHERE clause uses LIKE "%on" to filter surnames that end with "on".
  - The IN (0,1) condition ensures only valid active/inactive statuses are included.
- Example results show customers like Hanson (Inactive) and Gordon (Active).
- This helps in segmenting and analyzing specific customer groups based on naming patterns and activity status.

26. Can you observe any data disrupency in the Customer's data? As a hint it's present in the IsActiveMember and Exited columns. One more point to consider is that the data in the Exited Column is absolutely correct and accurate.

Query:

```
select * from bank_churn
where ActiveID = 1 and ExitID = 1;
SELECT
    COUNT(*) AS Total_Discrepancies
FROM bank_churn
WHERE ExitID = 1 AND ActiveID = 1;
```

Output:

Total_Discrepancies
735

Brief Answer Explanation

- This analysis identifies a data discrepancy in the bank_churn table concerning active status (ActiveID) and exit status (ExitID).
- According to the query, there are 735 records where ActiveID = 1 (marked as active) but ExitID = 1 (marked as exited).
- Since the Exited column is considered accurate, these records indicate a mismatch or error in the IsActiveMember/ActiveID column.
- Implication: Some customers are incorrectly marked as active despite having exited the bank.
- This discrepancy highlights the need for data cleaning or correction before performing analyses that rely on customer activity, such as churn analysis or retention strategy planning.