

Integrated Financial and Revenue Analysis

The objective of this project is to conduct a detailed analysis of financial performance and operational metrics using SQL queries to derive actionable insights. We aim to evaluate total policy revenue by product group, compare commissions and policy amounts across Account Executives, and analyze budget utilization versus new allocations. Additionally, we will assess the impact of meetings on opportunity closures, summarize quarterly revenue and transactions by Solution Group, identify top revenue-generating opportunities, and calculate average revenue per policy by Solution Group. By measuring revenue contributions of product groups and sub-groups, the project supports strategic decision-making and enhances financial and operational effectiveness.

[DATASET](#) || [SQL Script](#)

.....

□ Total Policy Revenue by Product Group for the Current Year:

- Write a query to calculate the total revenue from policies for each Product Group for the current year, and list the Product Groups in descending order of total revenue.

```
1  /* Total Policy Revenue by Product Group for the Current Year:
2   Write a query to calculate the total revenue from policies for each Product Group for the current year,
3   and list the Product Groups in descending order of total revenue. */
4
5  with currentyear as (
6   select year(policy_start_date) as yearS from brokerage
7   group by year(policy_start_date)
8   order by yearS desc
9   limit 1 )
10
11 select product_group , round(sum(b.amount),2) as TotalRevenue
12 from brokerage b
13 join currentyear cy
14 on year(b.policy_start_date) = cy.yearS
15 group by b.product_group
16 order by sum(amount) desc;
17
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
product_group	TotalRevenue		
▶ Marine	32991.55		
Miscellaneous	1390.13		
Employee Benefits	95.85		

□ Commission and Fees Analysis by Account Executive:

- Construct a query to compare the total commissions (from fees) and total policy amounts (from brokerage) for each Account Executive, and identify the Account Executive with the highest combined total.

```
19  /*Commission and Fees Analysis by Account Executive:
20  Construct a query to compare the total commissions (from fees) and total policy amounts (from brokerage) for each Account Executive,
21  and identify the Account Executive with the highest combined total*/
22
23  with TotalCommissions As (
24  select Account_Executive , sum(Amount) as TotalCommission from fees group by Account_Executive) ,
25  TotalPolicyAmount AS
26  (
27  select Exe_Name as Account_Executive , sum(Amount) as TotalPolicyAmt from brokerage group by Exe_Name
28  ),
29  CombinedCommission AS(
30  Select coalesce(c.Account_Executive,p.Account_Executive) AS Account_Executive,
31  coalesce(TotalCommission,0) + coalesce(TotalPolicyAmt,0) AS CombinedTotal
32  From TotalCommissions C
33  Left join TotalPolicyAmount p on c.Account_Executive = p.Account_Executive
34  UNION
35  Select coalesce(c.Account_Executive,p.Account_Executive) AS Account_Executive,
36  coalesce(TotalCommission,0) + coalesce(TotalPolicyAmt,0) AS CombinedTotal
37  From TotalCommissions C
38  right join TotalPolicyAmount p on c.Account_Executive = p.Account_Executive )
39  Select Account_Executive, Round(CombinedTotal,2) from CombinedCommission Order by CombinedTotal DESC Limit 1;
40
```

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

Account_Executive	Round(CombinedTotal,2)
Vinay	15441750.08

□ Budget Utilization vs. New Budget Allocation by Income Class:

- Develop a query to compare the actual budget utilization (from the fees table) against the new budget allocations (from the individual_budgets table) based on the income_class for each Account_Executive.

```
43  /* Budget Utilization vs. New Budget Allocation by Income Class:
44  Develop a query to compare the actual budget utilization (from the fees table) against the new budget allocations (from the individual_budgets table)
45  based on the income_class for each Account_Executive.
46  */
47
48  • With ActualBudgetUtilization AS (
49  Select Account_Executive , Sum(Amount) AS ActualBudget, income_class from fees group by Account_Executive, income_class),
50  NewBudgetAllocation AS (
51  Select Employee_Name as Account_Executive , Sum(New_Budget) AS new_budget,
52  'New' as income_class
53  from individual_budgets group by Employee_Name
54
55  UNION ALL
56
57  Select Employee_Name as Account_Executive , Sum(Cross_sell_budget) AS new_budget,
58  'Cross Sell' as income_class
59  from individual_budgets group by Employee_Name
60
61  UNION ALL
62
63  Select Employee_Name as Account_Executive , Sum(Renewal_Budget) AS new_budget,
64
65  'Renewal' as income_class
66  from individual_budgets group by Employee_Name
67  )
68
69  SELECT
70  a.Account_Executive AS Account_Executive,
71  a.income_class AS IncomeClass,
72  COALESCE(a.ActualBudget, 0) AS ActualUtilized,
73  COALESCE(b.new_budget, 0) AS BudgetAllocated
74  FROM ActualBudgetUtilization a
75  LEFT JOIN NewBudgetAllocation b
76  ON a.Account_Executive = b.Account_Executive
77  AND a.income_class = b.income_class
78  ORDER BY a.Account_Executive, a.income_class;
```

Result Grid Filter Rows: Export: Wrap Cell Content:				
	Account_Executive	IncomeClass	ActualUtilized	BudgetAllocated
▶	Abhinav Shivam	New	100000	129902
	Nishant Sharma	Cross Sell	396480	0
	Vinay	Renewal	18051	1500000

□ Impact of Meetings on Opportunity Closure:

- Create a query to evaluate the impact of meetings on the closure of opportunities by comparing the number of opportunities closed before and after each meeting, and identify meetings with the highest positive impact.

```
--
79  /* Impact of Meetings on Opportunity Closure:
80  Create a query to evaluate the impact of meetings on the closure of opportunities by comparing the number of opportunities closed before and after each meeting,
81  and identify meetings with the highest positive impact. */
82
83  WITH OpportunitiesBeforeAfter AS (
84  Select m.Account_Exe_Id AS Account_Exe_Id , m.Account_Executive AS Account_Executive, m.meeting_date AS meeting_date ,
85  Count(CASE when o.closing_date <= m.meeting_date THEN o.opportunity_id END) AS OpportunityBefore,
86  Count(CASE when o.closing_date > m.meeting_date THEN o.opportunity_id END) AS OpportunityAfter
87  from meeting m
88  left join opportunity o
89  on m.Account_Exe_Id = o. Account_Exe_Id
90  group by m.Account_Exe_Id , m.Account_Executive , m.meeting_date ),
91
92  ImpactAnalysis AS(
93  Select Account_Exe_Id, Account_Executive , meeting_date , OpportunityBefore, OpportunityAfter, (OpportunityAfter - OpportunityBefore) AS Impact
94  from OpportunitiesBeforeAfter group by Account_Exe_Id, Account_Executive , meeting_date )
95
96  select Account_Exe_Id, Account_Executive , meeting_date , OpportunityBefore, OpportunityAfter, Impact from ImpactAnalysis
97  Where Impact > 0
98  order by Impact DESC LIMIT 1 ;
99
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [F1](#)

	Account_Exe_Id	Account_Executive	meeting_date	OpportunityBefore	OpportunityAfter	Impact
▶	12	Shivani Sharma	2020-01-21	9	24	15

□ Quarterly Revenue and Transaction Summary by Solution Group:

- Write a SQL query to summarize the total revenue and count of invoices for each solution_group for each quarter in a given year. Your analysis should include:
 1. Total Revenue: Calculate the total revenue amount for each solution_group in each quarter of the specified year.
 2. Count of Invoices: Determine the count of invoices for each solution_group in each quarter of the specified year.

```
101 Write a SQL query to summarize the total revenue and count of invoices for each solution_group for each quarter in a given year.
102 Your analysis should include:
103 Total Revenue: Calculate the total revenue amount for each solution_group in each quarter of the specified year.
104 Count of Invoices: Determine the count of invoices for each solution_group in each quarter of the specified year. */
105
106 • SELECT
107     solution_group,
108     EXTRACT(QUARTER FROM invoice_date) AS quarter,
109     SUM(Amount) AS total_revenue,
110     COUNT(invoice_number) AS invoice_count
111 FROM invoice
112 WHERE
113     EXTRACT(YEAR FROM invoice_date) = 2019
114 GROUP BY
115     solution_group,
116     EXTRACT(QUARTER FROM invoice_date)
117 ORDER BY
118     solution_group, quarter desc;
119
```

	solution_group	quarter	total_revenue	invoice_count
▶	Construction, Power & Infrastructure	4	1156624	19
	Construction, Power & Infrastructure	3	175696	2
	Employee Benefits (EB)	4	9056	1
	Global Client Network (GNB Inward)	4	144725	5



□ Top 5 Opportunities by Revenue Amount:

- Construct a query to list the top 5 opportunities by their revenue amount, including the Account Executive responsible.

```
123  /* Top 5 Opportunities by Revenue Amount:
124  Construct a query to list the top 5 opportunities by their revenue amount, including the Account Executive responsible. */
125
126  • SELECT
127      opportunity_name,
128      opportunity_id,
129      Account_Executive,
130      revenue_amount
131  FROM
132      opportunity
133  ORDER BY
134      revenue_amount DESC
135  LIMIT 5;
136
```

	opportunity_name	opportunity_id	Account_Executive	revenue_amount
▶	Fire	OPP1900002070	Animesh Rawat	500000
	EL-Group Medicaid	OPP1900001042	Animesh Rawat	400000
	DB -Mega Policy	OPP1900001945	Ketan Jain	400000
	CVP GMC	OPP1900001365	Mark	350000
	FM-Group Medicaid	OPP1900001390	Mark	300000

□ Average Revenue per Policy by Solution Group

- Write a SQL query to calculate the average revenue per policy for each solution_group using the brokerage and invoice tables. Ensure that you consider only those policies that appear in both tables. List the solution groups with their average revenue per policy in descending order.

```
142  /* Average Revenue per Policy by Solution Group
143  Write a SQL query to calculate the average revenue per policy for each solution_group using the brokerage and invoice tables.
144  Ensure that you consider only those policies that appear in both tables. List the solution groups with their average revenue per policy
145  in descending order. */
146
147  WITH PolicyRevenue AS (
148      SELECT
149          b.solution_group,
150          COUNT(DISTINCT b.policy_number) AS policy_count,
151          SUM(i.Amount) AS total_revenue
152      FROM brokerage b JOIN invoice i
153      ON b.policy_number = i.policy_number GROUP BY b.solution_group )
154  SELECT
155      solution_group,
156      round((total_revenue / NULLIF(policy_count, 0)),0) AS average_revenue_per_policy
157  FROM PolicyRevenue ORDER BY average_revenue_per_policy DESC;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
solution_group		average_revenue_per_policy		
▶	Construction, Power & Infrastructure	2412730		
	Liability	59365		
	Global Client Network (GNB Inward)	24275		
	Marine	13566		
	Employee Benefits (EB)	9056		
	Property / BI	7767		
	Emerging Corporates Group (ECG)	2789		

□ Revenue Contribution by Product Sub-Group:

- What is the revenue contribution of each product sub-group as a percentage of total revenue?

```
159  /* Revenue Contribution by Product Sub-Group:
160  Question: What is the revenue contribution of each product sub-group as a percentage of total revenue? */
161
162  WITH TotalRevenue AS (
163      SELECT
164          SUM(revenue_amount) AS overall_revenue
165      FROM opportunity
166  )
167  SELECT
168      product_sub_group,
169      SUM(revenue_amount) AS sub_group_revenue,
170      round(((SUM(revenue_amount) * 1.0 / overall_revenue) * 100),2) AS revenue_percentage
171  FROM opportunity, TotalRevenue
172  GROUP BY product_sub_group, overall_revenue
173  ORDER BY revenue_percentage DESC;
174
```

	product_sub_group	sub_group_revenue	revenue_percentage
▶	Constructions & Infrastructure	2450000	35.62
	Medicaid	2315000	33.66
	Marine Hull	600000	8.72
	Miscellaneous	450000	6.54
	Engineering	329500	4.79
	Financial Lines	324000	4.71
	Political Risks	300000	4.36



□ Meeting Frequency Analysis

- How often do meetings occur per Account Executive, and how does this correlate with their revenue generation?

```
179 • WITH MeetingFrequency AS (  
180     SELECT Account_Exe_ID, Account_Executive, COUNT(*) AS meeting_count  
181     FROM meeting  
182     GROUP BY Account_Exe_ID, Account_Executive  
183 ),  
184 RevenueByExecutive AS (  
185     SELECT Account_Exe_ID, Account_Executive, SUM(Amount) AS total_revenue  
186     FROM  
187         (SELECT Account_Exe_ID, Exe_Name AS Account_Executive, Amount FROM brokerage  
188          UNION ALL  
189          SELECT Account_Exe_ID, Account_Executive, Amount FROM invoice  
190          UNION ALL  
191          SELECT Account_Exe_ID, Account_Executive, Amount FROM fees) AS combined_revenue  
192     GROUP BY Account_Exe_ID, Account_Executive )  
193 SELECT m.Account_Exe_ID, m.Account_Executive, m.meeting_count, round(r.total_revenue,0)  
194 FROM MeetingFrequency m  
195 LEFT JOIN RevenueByExecutive r  
196 ON m.Account_Exe_ID = r.Account_Exe_ID  
197 ORDER BY m.Account_Executive;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Account_Exe_ID	Account_Executive	meeting_count	round(r.total_revenue,0)
2	Abhinav Shivam	6	1941531
3	Animesh Rawat	3	7505554
3	Animesh Rawat	3	396480
4	Gilbert	2	386738

Conclusion

□ Detailed Financial and Operational Insights:

- The project provided a thorough analysis of policy revenue, commissions, and budget utilization, offering actionable insights into financial performance and operational metrics.

□ Performance Comparisons:

- Comparative analysis of commissions and policy amounts across Account Executives highlighted performance variations and areas for potential improvement.

□ Impact Analysis:

- Evaluated the effect of meetings on opportunity closures, enhancing understanding of sales strategies and their effectiveness.

□ Revenue and Transaction Trends:

- Summarized quarterly revenue and transactions by Solution Group and identified top revenue-generating opportunities, guiding focus on high-impact areas.

□ Strategic Decision Support:

- Measured revenue contributions of product groups and sub-groups to support strategic decision-making, ultimately improving financial and operational effectiveness.

.....