

```

SQLQuery1....\ACER (65)*  X  | sql athen aw...88\ACER (51)
1   SELECT COUNT(*) AS Total_Customers FROM churn_athena_ready;
2
3
4   SELECT TOP 10 * FROM churn_athena_ready;
5
6
7   SELECT
8       COLUMN_NAME,
9       DATA_TYPE,
10      CHARACTER_MAXIMUM_LENGTH
11  FROM INFORMATION_SCHEMA.COLUMNS
12 WHERE TABLE_NAME = 'churn_athena_ready';
13 GO
14

```

91 %    No issues found    Ln: 6, Ch: 1 | SPC | CRLF | Windows 1252

Results Messages

	Total_Customers														
1	4999														
1	customer_id	customer_uuid	age	gender	subscription_type	watch_hours	last_login_days	region	device	monthly_fee	churned	payment_method	number_of_profiles	avg_watch_	
2	1	49a5fdf9-7e69-4022-a6ad-0a1b9767fb5b	47	Other	Standard	0.699999988079071	19	Europe	Mobile	13.9899997711182	1	Gift Card	5	0.0299999	
3	2	4d71f6ce-fca9-4f7-8afa-197ac24de14b	27	Female	Standard	16.3199996948242	10	Asia	TV	13.9899997711182	0	Crypto	2	1.4800000	
4	3	d3c72c38-631b-4f9e-8a0e-de103cad1a7d	53	Other	Premium	4.51000022888184	12	Oceania	TV	17.9899997711182	1	Crypto	2	0.3499999	
5	4	4e265c34-103a-4dbb-9553-76c9aa47e946	56	Other	Standard	1.88999998569489	13	Africa	Mobile	13.9899997711182	1	Crypto	2	0.1299999	
6	5	d8079475-5be7-47e9-8782-ceb7ff61395e	58	Female	Standard	13.8000001907349	26	Oceania	Mobile	13.9899997711182	0	Debit Card	3	0.5099999	
7	6	8e63450a-13d6-4e83-bbb5-6aebde9152cb	48	Other	Basic	13.8299999237061	20	Asia	TV	8.98999977111816	0	Gift Card	5	0.6600000	
8	7	02387681-8c42-462a-807a-de0168c73b38	51	Male	Basic	14.3000001907349	56	Europe	Mobile	8.98999977111816	1	Gift Card	1	0.25	
9	8	0bcaad0c-545c-4ee1-85a6-75e165f39361	45	Other	Basic	9.97999954223633	10	Asia	Mobile	8.98999977111816	0	PayPal	3	0.9100000	

  

	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
1	customer_id	smallint	NULL
2	customer_uuid	nvarchar	50
3	age	tinyint	NULL
4	gender	nvarchar	50

Query executed successfully.

DESKTOP-MU3C888\SQLEXPRESS ... | DESKTOP-MU3C888\ACER (65) | ott\_churn db | 00:00:00 | Row: 1, Col: 1 | 31 rows

SQLQuery1...\ACER (65)\* X | sql athen aw...88\ACER (51))

```

4   SELECT
5     COUNT(*) AS total_customers,
6     SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned_customers,
7     CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10, 2)) AS churn_rate_percent,
8     -- Revenue Metrics
9     CAST(SUM(monthly_fee) AS DECIMAL(10, 2)) AS monthly_revenue,
10    CAST(SUM(annual_revenue) AS DECIMAL(10, 2)) AS annual_revenue,
11    CAST(SUM(annual_revenue) / 10000000.0 AS DECIMAL(10, 2)) AS annual_revenue_cr,
12    -- Loss Metrics
13    CAST(SUM(CASE WHEN churned = 1 THEN annual_revenue ELSE 0 END) AS DECIMAL(10, 2)) AS annual_loss,
14    CAST(SUM(CASE WHEN churned = 1 THEN annual_revenue ELSE 0 END) / 10000000.0 AS DECIMAL(10, 2)) AS loss_cr,
15    -- Risk Metrics
16    CAST(SUM(revenue_at_risk) AS DECIMAL(10, 2)) AS total_at_risk,
17    CAST(SUM(revenue_at_risk) / 10000000.0 AS DECIMAL(10, 2)) AS at_risk_cr,
18    -- Averages
19    CAST(AVG(monthly_fee) AS DECIMAL(10, 2)) AS avg_monthly_fee,
20    CAST(AVG(clv) AS DECIMAL(10, 2)) AS avg_clv,
21    CAST(AVG(risk_score) AS DECIMAL(10, 2)) AS avg_risk_score
22  FROM churn_athena_ready;
23  GO

```

91% v ① 1 ⚠ 0 ↑ ↓ Ln: 3, Ch: 35 SPC CRLF Windows 1252

Results Messages

	total_customers	churned_customers	churn_rate_percent	monthly_revenue	annual_revenue	annual_revenue_cr	annual_loss	loss_cr	total_at_risk	at_risk_cr	avg_monthly_fee	avg_clv	avg_risk_score
1	4999	2514	50.29	68408.01	820896.13	0.08	396010.32	0.04	621050.27	0.06	13.68	492.64	75.66

SQLQuery1...\ACER (65)\* X sql athen aw..88\ACER (51))

```

26 -- Q2: Risk Category Breakdown
27 =====
28
29     SELECT
30         risk_category,
31         COUNT(*) AS customer_count,
32         CAST(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS DECIMAL(10,2)) AS percentage,
33
34         -- Churn in each category
35         SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned_count,
36         CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
37
38         -- Financial Impact
39         CAST(SUM(annual_revenue) AS DECIMAL(10,2)) AS total_revenue,
40         CAST(SUM(revenue_at_risk) AS DECIMAL(10,2)) AS revenue_at_risk,
41         CAST(SUM(revenue_at_risk) / 100000.0 AS DECIMAL(10,2)) AS at_risk_lakhs,
42
43         -- Average Metrics
44         CAST(AVG(risk_score) AS DECIMAL(10,2)) AS avg_risk_score,
45         CAST(AVG(monthly_fee) AS DECIMAL(10,2)) AS avg_fee
46
47     FROM churn athena ready
48
49 1 ▲ 0 ↑ ↓ Ln: 3, Ch: 35 | SPC CRLF Windows 125

```

Results Messages

	total_customers	churned_customers	churn_rate_percent	monthly_revenue	annual_revenue	annual_revenue_cr	annual_loss	loss_cr	total_at_risk	at_risk_cr	avg_monthly_fee	avg_clv	avg_risk_score
1	4999	2514	50.29	60400.01	820096.13	0.08	396010.32	0.04	621050.27	0.06	13.68	492.64	75.66

	risk_category	customer_count	percentage	churned_count	churn_rate	total_revenue	revenue_at_risk	at_risk_lakhs	avg_risk_score	avg_fee
1	CRITICAL	774	15.48	694	69.66	120079.12	120435.50	1.20	94.02	13.79
2	HIGH	1857	37.15	1347	72.54	303161.16	249563.23	2.50	82.40	13.60
3	MEDIUM	1768	35.37	466	26.36	201903.84	197182.67	1.97	67.55	13.76
4	LOW	600	12.00	7	1.17	97752.00	53868.79	0.64	55.03	13.58

```

SQLQuery1....\ACER (65)*  X | sql athen aw...88\ACER (51)
1 -- =====
2 -- Q3: Subscription Type Performance
3 -- =====
4
5     SELECT
6         subscription_type,
7         COUNT(*) AS customers,
8         CAST(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS DECIMAL(10,2)) AS customer_percent,
9
10        -- Churn Metrics
11        SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned,
12        CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
13
14        -- Revenue
15        CAST(SUM(annual_revenue) AS DECIMAL(10,2)) AS total_revenue,
16        CAST(AVG(monthly_fee) AS DECIMAL(10,2)) AS avg_fee,
17        CAST(SUM(CASE WHEN churned = 1 THEN annual_revenue ELSE 0 END) AS DECIMAL(10,2)) AS loss_amount,
18
19        -- Engagement
20        CAST(AVG(watch_hours) AS DECIMAL(10,2)) AS avg_watch_hours,
21        CAST(AVG(last_login_days) AS DECIMAL(10,2)) AS avg_login_days,
22        CAST(AVG(avg_watch_time_per_day) AS DECIMAL(10,2)) AS avg_daily_watch
23
24    FROM churn_athena_ready
25    GROUP BY subscription_type
26    ORDER BY churn_rate DESC;
27
28

```

Ln: 25, Ch: 3 SPC CRLF Windows 1252

subscription_type	customers	customer_percent	churned	churn_rate	total_revenue	avg_fee	loss_amount	avg_watch_hours	avg_login_days	avg_daily_watch
Basic	1660	33.21	1026	61.81	179080.80	8.99	110684.88	11.52	30.00	0.92
Standard	1646	32.93	748	45.44	276330.40	13.00	125574.24	11.69	30.00	0.84
Premium	1693	33.87	740	43.71	365494.85	17.99	159751.20	11.72	30.00	0.86

```

SQLQuery1....\ACER (65)*  X  sql athen aw...88\ACER (51)
1   -- =====
2   -- Q4: Region-wise Analysis
3   -- =====
4   SELECT
5     region,
6     COUNT(*) AS customers,
7     CAST(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS DECIMAL(10,2)) AS customer_percent,
8
9     -- Churn
10    SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned_count,
11    CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
12
13     -- Revenue Impact
14    CAST(SUM(annual_revenue) / 100000.0 AS DECIMAL(10,2)) AS revenue_lakhs,
15    CAST(SUM(CASE WHEN churned = 1 THEN annual_revenue ELSE 0 END) / 100000.0 AS DECIMAL(10,2)) AS loss_lakhs,
16    CAST(SUM(revenue_at_risk) / 100000.0 AS DECIMAL(10,2)) AS at_risk_lakhs,
17
18     -- Engagement
19    CAST(AVG(watch_hours) AS DECIMAL(10,2)) AS avg_watch,
20    CAST(AVG(last_login_days) AS DECIMAL(10,2)) AS avg_login
21  FROM churn_athena_ready
22  GROUP BY region
23  ORDER BY loss_lakhs DESC;
24  GO

```

91 %    × 1 ⚠ 0 ↑ ↓

Results Messages

	region	customers	customer_percent	churned_count	churn_rate	revenue_lakhs	loss_lakhs	at_risk_lakhs	avg_watch	avg_login
1	Europe	867	17.34	448	51.67	1.42	0.71	1.08	11.41	30.00
2	South America	873	17.46	449	51.43	1.43	0.70	1.08	11.82	30.00
3	Asia	841	16.82	426	50.65	1.37	0.67	1.04	11.28	30.00
4	North America	851	17.02	421	49.47	1.39	0.66	1.06	11.91	30.00
5	Oceania	765	15.30	383	50.07	1.27	0.62	0.96	11.75	29.00
6	Africa	802	16.04	387	48.25	1.32	0.60	0.99	11.71	29.00

```

SQLQuery1....\ACER (65)*  X  sql athen aw...88\ACER (51)
1  -- =====
2  -- Q5: Payment Method Analysis
3  -- =====
4  SELECT
5      payment_method,
6      COUNT(*) AS users,
7      CAST(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS DECIMAL(10,2)) AS user_percent,
8
9      -- Risk Metrics
10     SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned,
11     CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
12     CAST(AVG(risk_score) AS DECIMAL(10,2)) AS avg_risk,
13
14     -- Financial
15     CAST(SUM(annual_revenue) / 1000000.0 AS DECIMAL(10,2)) AS revenue_lakhs,
16     CAST(SUM(CASE WHEN churned = 1 THEN annual_revenue ELSE 0 END) / 1000000.0 AS DECIMAL(10,2)) AS loss_lakhs,
17
18     -- Priority
19     CASE
20         WHEN AVG(churned * 1.0) > 0.4 THEN 'HIGH PRIORITY'
21         WHEN AVG(churned * 1.0) > 0.25 THEN 'MEDIUM PRIORITY'
22         ELSE 'LOW PRIORITY'
23     END AS action_priority
24
25     FROM churn_athena_ready
26     GROUP BY payment_method
27     ORDER BY churn_rate DESC;
    GO

```

91 %    × 1 ▲ 0 ↑ ↓

Ln: 2

Results Messages

	payment_method	users	user_percent	churned	churn_rate	avg_risk	revenue_lakhs	loss_lakhs	action_priority
1	Crypto	995	19.90	594	59.70	75.99	1.66	0.98	?? HIGH PRIORITY
2	Gift Card	975	19.50	563	57.74	76.18	1.61	0.91	?? HIGH PRIORITY
3	PayPal	1026	20.52	483	47.08	75.74	1.66	0.73	?? HIGH PRIORITY
4	Debit Card	1030	20.60	450	43.69	75.32	1.70	0.69	?? HIGH PRIORITY
5	Credit Card	973	19.46	424	43.58	75.09	1.58	0.64	?? HIGH PRIORITY

SQLQuery1....\ACER (65)\* X sql athen aw...88\ACER (51))

```
1  -- -----
2  -- Q6: Device-wise Churn Pattern
3  -- -----
4  SELECT
5      device,
6      COUNT(*) AS users,
7      CAST(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS DECIMAL(10,2)) AS user_percent,
8
9      -- Churn
10     SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned,
11     CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
12
13     -- Engagement
14     CAST(AVG(watch_hours) AS DECIMAL(10,2)) AS avg_watch,
15     CAST(AVG(avg_watch_time_per_day) AS DECIMAL(10,2)) AS avg_daily_watch,
16     CAST(AVG(last_login_days) AS DECIMAL(10,2)) AS avg_login,
17
18     -- Value
19     CAST(AVG(monthly_fee) AS DECIMAL(10,2)) AS avg_fee,
20     CAST(SUM(annual_revenue) / 1000000.0 AS DECIMAL(10,2)) AS revenue_lakhs
21
22 FROM churn_athena_ready
23 GROUP BY device
24 ORDER BY churn_rate DESC;
GO
```

91% ▾ × 1 ⚠ 0 ↑ ↓

Results Messages

	device	users	user_percent	churned	churn_rate	avg_watch	avg_daily_watch	avg_login	avg_fee	revenue_lakhs
1	Laptop	1006	20.12	521	51.79	11.28	0.82	29.00	13.67	1.65
2	Mobile	1004	20.08	507	50.50	11.98	0.89	30.00	13.61	1.64
3	Tablet	1048	20.96	524	50.00	10.77	0.81	29.00	13.63	1.71
4	TV	992	19.84	495	49.90	11.86	0.90	29.00	13.60	1.62
5	Desktop	949	18.98	467	49.21	12.41	0.96	30.00	13.92	1.59

SQLQuery1....\ACER (65)\* > X [sql athen aw..88\ACER (51)]

```
1  =====
2  -- Q7: Pareto Analysis - Top 20% Customers
3  =====
4  WITH customer_value AS (
5      SELECT
6          customer_id,
7          annual_revenue,
8          churned,
9          NTILE(5) OVER (ORDER BY annual_revenue DESC) AS value_rank
10     FROM churn_athena_ready
11 )
12     SELECT
13         CASE
14             WHEN value_rank = 1 THEN 'Top 20% (High Value)'
15             ELSE 'Bottom 80%'
16         END AS customer_segment,
17         COUNT(*) AS customer_count,
18         CAST(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS DECIMAL(10,2)) AS percentage,
19         CAST(SUM(annual_revenue) / 100000.0 AS DECIMAL(10,2)) AS total_revenue_lakhs,
20         SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned_count,
21         CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
22         CAST(SUM(CASE WHEN churned = 1 THEN annual_revenue ELSE 0 END) / 100000.0 AS DECIMAL(10,2)) AS loss_lakhs
23     FROM customer_value
24     GROUP BY
25         CASE
26             WHEN value_rank = 1 THEN 'Top 20% (High Value)'
27             ELSE 'Bottom 80%'
28         END;
29 GO
```

91% ▾ ② 1 ⚠ 0 ↑ ↓ Results Messages

Ln: 29, Ch: 3 SPC CRLF

	customer_segment	customer_count	percentage	total_revenue_lakhs	churned_count	churn_rate	loss_lakhs
1	??Top 20% (High Value)	1000	20.00	2.16	435	43.50	0.94
2	?? Bottom 80%	3999	80.00	6.05	2079	51.99	3.02

SQLQuery1....\ACER (65)\* X sql athen aw...88\ACER (51)

```
1  =====
2  -- Q9: Age Group Performance
3  =====
4
5  SELECT
6      CASE
7          WHEN age < 25 THEN 'Gen Z (<25)'
8          WHEN age BETWEEN 25 AND 34 THEN 'Millennials (25-34)'
9          WHEN age BETWEEN 35 AND 49 THEN 'Gen X (35-49)'
10         WHEN age >= 50 THEN 'Boomers (50+)'
11     END AS age_group,
12     COUNT(*) AS customers,
13     SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned,
14     CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
15     CAST(AVG(watch_hours) AS DECIMAL(10,2)) AS avg_watch,
16     CAST(AVG(monthly_fee) AS DECIMAL(10,2)) AS avg_fee,
17     CAST(SUM(annual_revenue) / 100000.0 AS DECIMAL(10,2)) AS revenue_lakhs
18 FROM churn_athena_ready
19 GROUP BY
20     CASE
21         WHEN age < 25 THEN 'Gen Z (<25)'
22         WHEN age BETWEEN 25 AND 34 THEN 'Millennials (25-34)'
23         WHEN age BETWEEN 35 AND 49 THEN 'Gen X (35-49)'
24         WHEN age >= 50 THEN 'Boomers (50+)'
25     END
26 ORDER BY churn_rate DESC;
GO
```

91 % ▾ × 1 ⚠ 0 ↑ ↓

Results Messages

	age_group	customers	churned	churn_rate	avg_watch	avg_fee	revenue_lakhs
1	Gen Z (<25)	707	358	50.64	10.94	13.80	1.17
2	Gen X (35-49)	1358	683	50.29	11.82	13.58	2.21
3	Millennials (25-34)	947	476	50.26	11.30	13.76	1.56
4	Boomers (50+)	1987	997	50.18	11.94	13.68	3.26

```
1 -- =====
2 -- Q10: Genre Preferences & Churn
3 -- =====
4
5 SELECT
6     favorite_genre,
7     COUNT(*) AS customers,
8     SUM(CASE WHEN churned = 1 THEN 1 ELSE 0 END) AS churned,
9     CAST(AVG(churned * 1.0) * 100 AS DECIMAL(10,2)) AS churn_rate,
10    CAST(AVG(watch_hours) AS DECIMAL(10,2)) AS avg_watch,
11    CAST(AVG(avg_watch_time_per_day) AS DECIMAL(10,2)) AS avg_daily_watch,
12    CAST(AVG(monthly_fee) AS DECIMAL(10,2)) AS avg_fee
13 FROM churn_athena_ready
14 GROUP BY favorite_genre
15 ORDER BY churn_rate DESC;
GO
```

91% ▾ × 1 ⚠ 0 ↑ ↓

Results Messages

	favorite_genre	customers	churned	churn_rate	avg_watch	avg_daily_watch	avg_fee
1	Action	696	364	52.30	10.91	0.78	13.48
2	Drama	731	382	52.26	11.64	0.88	13.84
3	Horror	713	367	51.47	11.05	0.68	13.82
4	Documentary	729	370	50.75	11.48	0.95	13.78
5	Comedy	685	342	49.93	12.13	0.81	13.55
6	Romance	725	350	48.28	12.66	1.05	13.50
7	Sci-Fi	720	339	47.08	11.62	0.96	13.81

Query executed successfully.

DESKTOP-MU3C888\SOLEXPRESS ... DESKTOP-MU3C888\ACER

SQLQuery1....\ACER (65)\* X sql athen aw...88\ACER (51)

```
1 --- =====
2 -- Q11: Key Drivers of Churn
3 --- =====
4 SELECT
5     'Last Login Days' AS factor,
6     AVG(CASE WHEN churned = 1 THEN last_login_days ELSE NULL END) AS avg_churned,
7     AVG(CASE WHEN churned = 0 THEN last_login_days ELSE NULL END) AS avg_retained,
8     AVG(CASE WHEN churned = 1 THEN last_login_days ELSE NULL END) -
9     AVG(CASE WHEN churned = 0 THEN last_login_days ELSE NULL END) AS impact
10    FROM churn_athena_ready
11 UNION ALL
12 SELECT
13     'Watch Hours',
14     AVG(CASE WHEN churned = 1 THEN watch_hours ELSE NULL END),
15     AVG(CASE WHEN churned = 0 THEN watch_hours ELSE NULL END),
16     AVG(CASE WHEN churned = 1 THEN watch_hours ELSE NULL END) -
17     AVG(CASE WHEN churned = 0 THEN watch_hours ELSE NULL END)
18    FROM churn_athena_ready
19 UNION ALL
20 SELECT
21     'Monthly Fee',
22     AVG(CASE WHEN churned = 1 THEN monthly_fee ELSE NULL END),
23     AVG(CASE WHEN churned = 0 THEN monthly_fee ELSE NULL END),
24     AVG(CASE WHEN churned = 1 THEN monthly_fee ELSE NULL END) -
25     AVG(CASE WHEN churned = 0 THEN monthly_fee ELSE NULL END)
26    FROM churn_athena_ready;
27 GO
```

91% ▾ × 1 ⚠ 0 ↑ ↓

Results Messages

	factor	avg_churned	avg_retained	impact
1	Last Login Days	38	21	17
2	Watch Hours	5.90376957204598	17.4509464000389	-11.5471768279929
3	Monthly Fee	13.126833502224	14.2483498717218	-1.12151636949781

SQLQuery1....\ACER (65)\* sql athen aw...88\ACER (51))

```
26 UNION ALL
27
28
29     SELECT
30         'Payment Method Incentive',
31         'payment_method = ''Gift Card''' ,
32         COUNT(*),
33         CAST(SUM(annual_revenue) AS DECIMAL(10,2)),
34         CAST(SUM(annual_revenue) * 0.2 AS DECIMAL(10,2)),
35         CAST(COUNT(*) * 100.0 AS DECIMAL(10,2))
36     FROM churn_athena_ready
37     WHERE payment_method = 'Gift Card' AND churned = 0
38
39 )
40     SELECT
41         campaign,
42         target,
43         target_customers,
44         CAST(total_value / 100000.0 AS DECIMAL(10,2)) AS total_value_lakhs,
45         CAST(potential_savings / 100000.0 AS DECIMAL(10,2)) AS savings_lakhs,
46         CAST(campaign_cost / 100000.0 AS DECIMAL(10,2)) AS cost_lakhs,
47         CAST((potential_savings - campaign_cost) / campaign_cost * 100 AS DECIMAL(10,2)) AS roi_percent,
48         CASE
49             WHEN (potential_savings - campaign_cost) / campaign_cost * 100 > 500 THEN 'HIGH PRIORITY'
50             WHEN (potential_savings - campaign_cost) / campaign_cost * 100 > 300 THEN 'MEDIUM PRIORITY'
51             ELSE 'LOW PRIORITY'
52         END AS priority
53     FROM campaigns
54     ORDER BY roi_percent DESC;
GO|
```

91% ▾ × 1 ⚠ 0 ↑ ↓

Results Messages

	campaign	target	target_customers	total_value_lakhs	savings_lakhs	cost_lakhs	roi_percent	priority
1	Payment Method Incentive	payment_method = 'Gift Card'	412	0.70	0.14	0.41	-65.91	?? LOW PRIORITY
2	Engagement Boost	watch_hours < 5	281	0.53	0.13	0.42	-68.85	?? LOW PRIORITY
3	Win-back Campaign	last_login_days > 30	615	1.09	0.33	1.23	-73.40	?? LOW PRIORITY