QueryForge

CRM Analytics Metrics Reference

50+ Custom Metrics for your crm.db Snowflake Schema

50+	6	18	21,397+
Metrics Defined	Categories	Tables Covered	Records in crm.db

TABLE OF CONTENTS

01. SALES PERFORMANCE

12 metrics in this section | IDs: **SP-01** — **SP-12**

SP-01 Revenue

Total Revenue

Sum of all net sale amounts for completed transactions. The primary top-line revenue metric.

SQL: SUM(net_amount) WHERE order_status = 'Completed'

Tables: **fact_sales**

SP-02 Revenue

Gross Revenue

Total revenue before any discounts are applied. Shows the ceiling of what could have been earned.

SQL: SUM(gross_amount) WHERE order_status = 'Completed'

Tables: **fact_sales**

SP-03 Revenue

Total Discount Given

Aggregate discount value surrendered across all transactions. High values indicate pricing pressure.

SQL: SUM(discount_amount) WHERE order_status = 'Completed'

Tables: **fact_sales**

SP-04 Pricing

Average Discount Rate

Mean discount percentage applied per transaction. Benchmarks negotiation aggressiveness across the sales team.

SQL: AVG(discount_pct) WHERE order_status = 'Completed'

Tables: **fact_sales**

SP-05 Profitability

Gross Profit

Revenue remaining after subtracting the cost of goods sold. Core profitability indicator.

SQL: SUM(gross_profit) WHERE order_status = 'Completed'

Tables: **fact_sales**

SP-06 Profitability

Gross Profit Margin %

Percentage of revenue retained as profit after COGS. Key indicator of product and pricing health.

*SQL: SUM(gross_profit) / SUM(net_amount) * 100*

Tables: **fact_sales**

SP-07 Revenue

Average Order Value (AOV)

Mean revenue per completed transaction. Rising AOV indicates upsell success or product mix shift.

SQL: SUM(net_amount) / COUNT(sale_id) WHERE order_status = 'Completed'

Tables: **fact_sales**

SP-08 Pricing

Revenue per Unit Sold

Average net revenue generated per individual unit. Useful for comparing product line efficiency.

SQL: SUM(net_amount) / SUM(quantity)

Tables: **fact_sales**

SP-09 Growth

Monthly Revenue Growth Rate

Month-over-month percentage change in net revenue. Tracks sales momentum and seasonal patterns.

*SQL: (This Month Revenue - Last Month Revenue) / Last Month Revenue * 100*

Tables: **fact_sales, dim_date**

SP-10 Growth

Quarter-over-Quarter Revenue

Compares revenue of the current quarter against the prior quarter to assess growth trajectory.

SQL: SUM(net_amount) GROUP BY quarter, year

Tables: **fact_sales, dim_date**

SP-11 Quality

Cancellation Rate

Percentage of orders that were cancelled before completion. High rates signal fulfilment or CX issues.

*SQL: COUNT(*) WHERE order_status = 'Cancelled' / COUNT(*) * 100*

Tables: **fact_sales**

SP-12 Quality

Refund Rate

Percentage of completed transactions that resulted in a refund. Indicates product or expectation mismatches.

*SQL: COUNT(*) WHERE order_status = 'Refunded' / COUNT(*) * 100*

Tables: **fact_sales**

02. SALESPERSON METRICS

8 metrics in this section | IDs: **SM-01 — SM-08**

SM-01 Performance

Revenue per Salesperson

Total net revenue attributed to each individual sales rep. Primary leaderboard metric.

SQL: SUM(net_amount) GROUP BY salesperson_id

Tables: **fact_sales, dim_salesperson**

SM-02 Performance

Quota Attainment %

Percentage of individual revenue quota achieved. Core KPI for sales compensation and performance reviews.

*SQL: SUM(net_amount) / quota * 100 GROUP BY salesperson_id*

Tables: **fact_sales, dim_salesperson**

SM-03 Activity

Deals Closed per Rep

Count of completed transactions per salesperson. Measures activity volume independent of deal size.

SQL: COUNT(sale_id) WHERE order_status = 'Completed' GROUP BY salesperson_id

Tables: **fact_sales, dim_salesperson**

SM-04 Performance

Average Deal Size per Rep

Mean order value for each salesperson. Identifies reps who consistently win larger deals.

SQL: SUM(net_amount) / COUNT(sale_id) GROUP BY salesperson_id

Tables: **fact_sales, dim_salesperson**

SM-05 Pricing

Discount Rate per Rep

Average discount given by each rep. Outliers indicate either strong negotiators or pricing compliance issues.

SQL: AVG(discount_pct) GROUP BY salesperson_id

Tables: **fact_sales, dim_salesperson**

SM-06 Activity

Activities per Rep

Total logged calls, emails, meetings, and demos per salesperson. Measures top-of-funnel effort.

SQL: COUNT(activity_id) GROUP BY salesperson_id

Tables: **crm_activities, dim_salesperson**

SM-07 Efficiency

Revenue per Activity

How much revenue each logged activity generates on average. Measures sales efficiency per effort unit.

SQL: SUM(net_amount) / COUNT(activity_id) JOIN by salesperson_id

Tables: **fact_sales, crm_activities**

SM-08 Performance

Win Rate per Rep

Percentage of opportunities closed as won versus total closed opportunities per salesperson.

*SQL: COUNT(*) WHERE stage = 'Closed Won' / COUNT(*) WHERE stage LIKE 'Closed%' * 100*

Tables: **crm_opportunities, dim_salesperson**

03. PIPELINE & OPPORTUNITY METRICS

9 metrics in this section | IDs: **PM-01** — **PM-09**

PM-01 Pipeline

Total Pipeline Value

Sum of expected values across all open opportunities. Represents the maximum potential revenue in play.

SQL: SUM(expected_value) WHERE stage NOT IN ('Closed Won','Closed Lost')

Tables: **crm_opportunities**

PM-02 Forecast

Weighted Pipeline Value

Pipeline value adjusted by close probability per stage. More realistic revenue forecast than raw pipeline.

*SQL: SUM(expected_value * probability / 100)*

Tables: **crm_opportunities**

PM-03 Performance

Win Rate

Overall percentage of opportunities that close as won. Fundamental measure of sales team effectiveness.

*SQL: COUNT(*) WHERE stage = 'Closed Won' / COUNT(*) WHERE stage LIKE 'Closed%' * 100*

Tables: **crm_opportunities**

PM-04 Velocity

Average Deal Cycle Length

Mean number of days from opportunity creation to close. Long cycles indicate friction or complexity.

SQL: AVG(julianday(actual_close) - julianday(created_at)) WHERE stage = 'Closed Won'

Tables: **crm_opportunities**

PM-05 Pipeline

Pipeline Stage Distribution

Count and value of opportunities at each stage. Reveals funnel shape and where deals are stalling.

SQL: COUNT(*), SUM(expected_value) GROUP BY stage

Tables: **crm_opportunities**

PM-06 Pipeline

Average Expected Deal Value

Mean expected value per open opportunity. Indicates the calibre of deals currently in the pipeline.

SQL: AVG(expected_value) WHERE stage NOT IN ('Closed Won','Closed Lost')

Tables: **crm_opportunities**

PM-07 Analysis

Lost Deal Rate by Stage

Where in the funnel deals are most commonly lost. Identifies specific stages with conversion problems.

SQL: COUNT(*) WHERE stage = 'Closed Lost' GROUP BY previous_stage

Tables: **crm_opportunities**

PM-08 Forecast

Revenue Realisation Rate

Ratio of actual closed value to expected value. Values below 1.0 mean deals close smaller than forecast.

SQL: SUM(actual_value) / SUM(expected_value) WHERE stage = 'Closed Won'

Tables: **crm_opportunities**

PM-09 Funnel

Opportunity Conversion Rate

Percentage of leads that convert to a formal opportunity. Measures lead qualification effectiveness.

*SQL: COUNT(DISTINCT opportunity_id) / COUNT(DISTINCT lead_id) * 100*

Tables: **crm_opportunities, crm_leads**

04. LEAD & FUNNEL METRICS

8 metrics in this section | IDs: **LF-01** — **LF-08**

LF-01 Funnel

Lead Conversion Rate

Percentage of total leads that convert to opportunities or customers. Core top-of-funnel health metric.

*SQL: COUNT(*) WHERE status = 'Converted' / COUNT(*) * 100*

Tables: **crm_leads**

LF-02 Acquisition

Leads by Source

Distribution of lead volume across acquisition channels. Identifies which channels produce the most leads.

SQL: COUNT(*) GROUP BY source

Tables: **crm_leads**

LF-03 Acquisition

Lead Value by Source

Total estimated value of leads grouped by source. Identifies highest-value acquisition channels.

SQL: SUM(estimated_value) GROUP BY source

Tables: **crm_leads**

LF-04 Acquisition

Average Lead Value

Mean estimated value per lead. Useful for calculating return on marketing investment per channel.

SQL: AVG(estimated_value) GROUP BY source

Tables: **crm_leads**

LF-05 Velocity

Lead Response Time

Average time between lead creation and first logged activity. Faster response correlates with higher conversion.

SQL: AVG(julianday(first_activity_date) - julianday(lead_created_at))

Tables: **crm_leads, crm_activities**

LF-06 Quality

Qualified Lead Rate

Percentage of leads that reach 'Qualified' status. Measures how well the team identifies good-fit prospects.

*SQL: COUNT(*) WHERE status = 'Qualified' / COUNT(*) * 100*

Tables: **crm_leads**

LF-07 Quality

Lead Loss Rate

Percentage of leads marked as lost or unqualified. High rates may indicate poor targeting or messaging.

*SQL: COUNT(*) WHERE status IN ('Lost','Unqualified') / COUNT(*) * 100*

Tables: **crm_leads**

LF-08 Geographic

Leads by Geography

Volume and value of leads grouped by region and country. Reveals geographic market opportunities.

SQL: COUNT(*), SUM(estimated_value) GROUP BY region_id

Tables: **crm_leads, dim_geography, dim_region**

05. CUSTOMER METRICS

8 metrics in this section | IDs: **CM-01** — **CM-08**

CM-01 Segmentation

Revenue by Customer Segment

Net revenue broken down by Enterprise, Mid-Market, SMB, Startup, and Government segments.

SQL: SUM(net_amount) GROUP BY segment_name

Tables: **fact_sales, dim_customer, dim_segment**

CM-02 Account

Top 10 Customers by Revenue

Ranked list of highest-spending customers. Identifies key accounts that need retention focus.

SQL: SUM(net_amount) GROUP BY customer_id ORDER BY SUM DESC LIMIT 10

Tables: **fact_sales, dim_customer**

CM-03 Behaviour

Customer Purchase Frequency

Average number of transactions per active customer. Low frequency signals churn or single-purchase behaviour.

SQL: COUNT(sale_id) / COUNT(DISTINCT customer_id)

Tables: **fact_sales**

CM-04 Value

Average Revenue per Customer

Mean lifetime revenue per unique customer. Simple proxy for customer value without full LTV modelling.

SQL: SUM(net_amount) / COUNT(DISTINCT customer_id)

Tables: **fact_sales, dim_customer**

CM-05 Segmentation

Revenue by Industry

Net revenue grouped by customer industry vertical. Identifies strongest and weakest industry segments.

SQL: SUM(net_amount) GROUP BY industry

Tables: **fact_sales, dim_customer**

CM-06 Retention

New vs Returning Customer Revenue

Revenue split between first-time buyers and repeat customers. Measures customer loyalty and retention quality.

SQL: CASE WHEN first purchase = current purchase THEN 'New' ELSE 'Returning' END

Tables: **fact_sales, dim_customer**

CM-07 Risk

Customers with Open Support Tickets

Count of active customers who currently have unresolved support tickets. Flags churn risk accounts.

SQL: COUNT(DISTINCT customer_id) WHERE ticket status IN ('Open','In Progress','Escalated')

Tables: **crm_support_tickets, dim_customer**

CM-08 Geographic

Revenue by Geography

Net revenue grouped by city, country, and region. Essential for territory planning and resource allocation.

SQL: SUM(net_amount) GROUP BY country_id, region_id

Tables: **fact_sales, dim_customer, dim_geography**

06. PRODUCT METRICS

7 metrics in this section | IDs: **PRD-01** — **PRD-07**

PRD-01 Product

Revenue by Product

Net revenue per product SKU. Identifies best-sellers and underperforming items in the catalogue.

SQL: SUM(net_amount) GROUP BY product_id ORDER BY SUM DESC

Tables: **fact_sales, dim_product**

PRD-02 Volume

Units Sold per Product

Total quantity sold per product. Volume metric independent of price — useful for inventory planning.

SQL: SUM(quantity) GROUP BY product_id

Tables: **fact_sales, dim_product**

PRD-03 Category

Revenue by Category

Net revenue aggregated by product category (Software, Hardware, Services, Cloud).

SQL: SUM(net_amount) GROUP BY category_name

Tables: **fact_sales, dim_product, dim_category**

PRD-04 Profitability

Profit Margin by Product

Gross profit as a percentage of net revenue per product. Reveals true profitability of each SKU.

*SQL: SUM(gross_profit) / SUM(net_amount) * 100 GROUP BY product_id*

Tables: **fact_sales, dim_product**

PRD-05 Brand

Revenue by Brand

Net revenue broken down by brand. Useful for brand portfolio and licensing analysis.

SQL: SUM(net_amount) GROUP BY brand_name

Tables: **fact_sales, dim_product, dim_brand**

PRD-06 Pricing

Average Selling Price vs List Price

Compares the actual unit price at which products sell versus the listed price in dim_product.

SQL: AVG(f.unit_price) vs p.unit_price GROUP BY product_id

Tables: **fact_sales, dim_product**

PRD-07 Profitability

Top Products by Profit Contribution

Ranks products by absolute gross profit generated, not just revenue. True value-drivers ranking.

SQL: SUM(gross_profit) GROUP BY product_id ORDER BY SUM DESC

Tables: **fact_sales, dim_product**

07. CHANNEL & SUPPORT METRICS

12 metrics in this section | IDs: **CH-01** — **ACT-03**

CH-01 Channel

Revenue by Channel

Net revenue grouped by sales channel (Direct, Online, Partner, Phone, Referral).

SQL: SUM(net_amount) GROUP BY channel_name

Tables: **fact_sales, dim_channel**

CH-02 Channel

Average Order Value by Channel

Mean deal size per channel. Direct channels typically yield larger deals than online or referral.

SQL: SUM(net_amount) / COUNT(sale_id) GROUP BY channel_name

Tables: **fact_sales, dim_channel**

CH-03 Channel

Channel Mix %

Percentage of total revenue contributed by each channel. Monitors over-dependence on any single route to market.

*SQL: SUM(net_amount) / total_revenue * 100 GROUP BY channel_name*

Tables: **fact_sales, dim_channel**

SU-01 Support

Ticket Volume by Status

Count of support tickets in each status (Open, In Progress, Resolved, Closed, Escalated).

SQL: COUNT(*) GROUP BY status

Tables: **crm_support_tickets**

SU-02 Support

Average Ticket Resolution Time

Mean hours or days between ticket creation and resolution. Key SLA compliance metric.

SQL: AVG(julianday(resolved_at) - julianday(created_at)) WHERE resolved_at IS NOT NULL

Tables: **crm_support_tickets**

SU-03 Support

Ticket Volume by Priority

Distribution of tickets across Low, Medium, High, and Critical priorities. Tracks operational urgency.

SQL: COUNT(*) GROUP BY priority

Tables: **crm_support_tickets**

SU-04 Support

Escalation Rate

Percentage of tickets that escalate to critical status. High rates indicate systemic product or process issues.

*SQL: COUNT(*) WHERE status = 'Escalated' / COUNT(*) * 100*

Tables: **crm_support_tickets**

SU-05 Support

Support Tickets per Customer

Average number of tickets raised per customer. High values flag problematic accounts or products.

SQL: COUNT(*) / COUNT(DISTINCT customer_id)

Tables: **crm_support_tickets**

SU-06 Support

Tickets by Product

Volume of support tickets associated with each product. Surfaces quality issues in specific SKUs.

SQL: COUNT(*) GROUP BY product_id

Tables: **crm_support_tickets, dim_product**

ACT-01 Activity

Activity Volume by Type

Count of each activity type (Call, Email, Meeting, Demo, Follow-up). Shows team engagement patterns.

SQL: COUNT(*) GROUP BY activity_type

Tables: **crm_activities**

ACT-02 Activity

Activity Outcome Distribution

How activities resolve — Positive, Neutral, Negative, No Answer, Converted. Quality of outreach indicator.

SQL: COUNT(*) GROUP BY outcome

Tables: **crm_activities**

ACT-03 Efficiency

Activities Leading to Conversion

Average number of activities logged before a lead converts. Helps optimise outreach cadence.

SQL: AVG(activity_count) WHERE lead status changes to 'Converted'

Tables: **crm_activities, crm_leads**