**Sales Conversion Funnel Optimization**

## **Problem Statement:**

The sales team lacks a structured, data-driven approach to monitor and optimize their lead-to-deal conversion funnel. Without clear visibility into stage-level performance, drop-offs, and conversion inefficiencies, they are unable to identify bottlenecks, evaluate representative or regional performance, or make informed strategic decisions. This limits their ability to improve conversion rates, shorten sales cycles, and prioritize high-impact leads effectively.

## **Project Objective:**

To create a dynamic analytics solution that integrates CRM data to:

* Map the lead-to-deal sales funnel
* Track and visualize funnel performance metrics
* Identify high-dropoff points and regional/reps variances
* Automate sales reporting for business stakeholders
* Generate actionable recommendations for conversion improvement

# **Week 1: Planning, Setup & Data Source Integration**

## **Objectives:**

* Finalize project scope, define KPIs
* Divide team roles
* Identify CRM data sources and ingestion strategy

## **Tasks:**

* Define funnel: Lead → Contacted → Demo → Proposal → Deal/Lost
* Choose CRM source (e.g., mock CSV, HubSpot export)
* Decide ingestion mode: manual upload, Google Sheets API, or App Script
* Draft CRM schema
* Set up Google Sheets + App Script base ingestion

## **Deliverables:**

* Funnel stage definitions
* Project plan (roles, tools, timeline)
* Initial CRM data files (mock/simulated)
* Data ingestion script prototype
* Schema design document

## **Project Scope :**

Build a dynamic, data-driven sales funnel analytics solution using CRM data that enables the sales team to track, analyze, and improve their lead-to-deal conversion performance.

### **Scope Includes:**

* Mapping the lead-to-deal sales funnel
* Tracking stage-wise performance metrics (conversion rate, drop-offs, etc.)
* Identifying regional & rep-level variations
* Automating CRM data ingestion (Google Sheets/App Script or API)
* Visualizing funnel and KPIs
* Generating actionable recommendations
* Delivering weekly updates and final report/dashboard

## **Funnel Stages**

| **Stage Name** | **Description** |
| --- | --- |
| **Lead Captured** | A new lead enters the CRM from any source (e.g., form fill, referral, ad) |
| **Lead Qualified** | Lead passes basic filters (e.g., correct industry, budget, role) – MQL |
| **Contacted** | Sales rep initiates outreach (call, email, LinkedIn, etc.) |
| **Responded (optional)** | Lead replies to rep — useful to track follow-up effectiveness |
| **Demo Scheduled** | Lead agrees to book a product demo or sales meeting |
| **Demo Completed (optional)** | Lead attends the demo — not all scheduled demos happen! |
| **Proposal Sent** | Proposal or pricing document shared with lead |
| **Negotiation (optional)** | Lead engages in pricing/contract discussions |
| **Deal Won** | Lead accepts the proposal and signs the contract |
| **Deal Lost** | Lead rejects the proposal, goes silent, or is disqualified |

**Deal Source Values (as Enum):**

| **Source Type** | **Meaning** |
| --- | --- |
| **Inbound** | The customer found you (e.g., via your website, SEO, blog, etc.) |
| **Outbound** | Your sales team reached out (e.g., cold email, cold call) |
| **Referral** | Referred by another client, employee, or partner |
| **Event** | Captured during a conference, webinar, or offline event |
| **Paid Ads** | Google Ads, Facebook Ads, LinkedIn Ads, etc. |
| **Partner** | Through a reseller, agency, or channel partner |
| **Web Form** | Directly filled a contact/sales form on your website |
| **Re-engagement** | A previously lost lead that re-engaged |

## **Team Roles and Responsibilities for week 1**

**Anurag : Project Scope + Funnel Logic**

| **Responsibilities** |
| --- |

* Finalize **project scope** and objective
* Define **funnel stages**:  
   Lead → Contacted → Demo → Proposal → Deal/Lost
* Document the funnel logic clearly
* Contribute to team role division

**Aman : KPI Definition + Reporting Plan**

| **Responsibilities** |
| --- |

* Define **KPIs** (e.g., conversion rate, drop-off rate)
* Describe **how KPIs will be calculated**
* Draft initial **reporting structure** (rep-wise, region-wise)
* Coordinate with schema designer to ensure KPI fields are tracked

**Chandini : Schema Design + Google Sheet Setup**

| **Responsibilities** |
| --- |

* Design CRM **data schema** based on tables: Lead, Sales\_Rep, Region, Stage\_History, Deal, Company
* Create **Google Sheet** with tabs for each table
* Add proper column headers and formats
* Set up data validation (e.g., dropdowns for stage, status)

**Hamsapriya : Mock Data Generation**

| **Responsibilities** |
| --- |

* Use **Python + Random** to generate **mock CRM data**
* Create .csv files for each table (lead.csv, rep.csv, etc.)
* Ensure data is realistic and matches schema
* Coordinate with ingestion team to upload/import to Sheets

**Bala Ramana : Ingestion Script + Documentation**

| **Responsibilities** |
| --- |

* Set up **Google Apps Script** to insert mock data into the Lead tab
* Automate new lead entry
* Write **documentation** for how the script works

## 

## **KPIs and Formulae**

### **Primary KPI’s**

**1. Conversion Rate per Funnel Stage**

% of leads that move from one stage to the next (e.g., Contacted → Demo)

**Schema:**

* **Table:** Stage\_History
* **Fields:** Lead\_ID, Stage

**Formula (Conceptual):**

Conversion\_Rate = COUNT(Leads with Stage = "Demo") / COUNT(Leads with Stage = "Contacted")

**Sheets Formula:**

=COUNTIFS(Stage\_History!C:C, "Demo") / COUNTIFS(Stage\_History!C:C, "Contacted") \* 100

**2. Drop-off % at Each Stage**

% of leads that don't progress to the next stage

**Schema:**

* **Table:** Stage\_History
* **Fields:** Lead\_ID, Stage

**Formula (Conceptual):**

Drop\_Off = (Leads in Stage N but not in N+1) / (Leads in Stage N)

**Sheets Formula:**

=(COUNTIFS(Stage\_History!C:C, "Contacted") - COUNTIFS(Stage\_History!C:C, "Demo")) / COUNTIFS(Stage\_History!C:C, "Contacted") \* 100

### **Secondary KPI’s**

**3. Regional / Rep Performance**

Conversion or deal count by rep or region

**Schema:**

* **Tables:** Lead, Sales\_Rep, Region, Deal
* **Fields:**
  + Lead.Rep\_ID, Lead.Region\_ID, Deal.Status, Deal.Amount
  + Joined via Rep\_ID, Region\_ID

**Sheets Formula:**

=COUNTIFS(Deal!C:C, "Won", Lead!E:E, "Rep\_123") ← for one rep

Best to use a **pivot table**:

* Rows: Rep\_Name
* Values: Count of Deals where Status = "Won"

**4. Follow-up Response Rate**

% of contacted leads that responded

**Schema:**

* **Table:** Stage\_History
* **Fields:** Lead\_ID, Stage

**Formula:**

=COUNTIFS(Stage\_History!C:C, "Responded") / COUNTIFS(Stage\_History!C:C, "Contacted") \* 100

**5. Proposal Acceptance Rate**

% of proposals that turned into deals

**Schema:**

* **Tables:** Stage\_History, Deal
* **Fields:** Stage, Deal.Status, Deal.Lead\_ID

**Formula:**

=COUNTIFS(Deal!C:C, "Won") / COUNTIFS(Stage\_History!C:C, "Proposal") \* 100

**6. Cold Lead Re-entry Rate**

% of leads marked "Lost" that re-enter the funnel

**Schema:**

* **Tables:** Deal, Stage\_History
* **Fields:** Deal.Status = Lost, then look for later Stage\_History entries for that same Lead\_ID

**Logic:**

1. Identify Lead\_IDs in Deal.Status = "Lost"
2. Check if those IDs appear again in Stage\_History with a **later Timestamp** and Stage in (Lead, Contacted, etc.)

It will be best with:

* SQL with MAX(Timestamp) and join
* OR a filtered VLOOKUP in Sheets

### **Independent KPIs**

**7. Lead Response Time**

How quickly a rep responds to a new lead

**Schema:**

* From Lead.Created\_Date and first "Contacted" stage timestamp in Stage\_History

**Formula:**

Avg Response Time = Avg(First\_Contact\_Timestamp - Lead\_Created\_Date)

**8. Deal Win Rate**

% of total leads that result in a successful deal

**Schema:**

* From Deal.Status and Lead\_ID

**Formula:**

Deal Win Rate = (Deals Won) / (Total Leads) × 100

**9. Lead Source Performance**

Which source brings the most valuable or high-converting leads

**Schema:**

* Lead.Source, Lead\_ID, Deal.Status, Deal.Amount

**Formula:**

Conversion Rate by Source = (Deals Won from Source) / (Total Leads from Source)

or

Avg Deal Value by Source = SUM(Amount) / COUNT(Source)

**10. First Contact Rate**

% of leads that were actually contacted

**Schema:**

* Stage\_History.Stage = "Contacted"

**Formula:**

= COUNT(Leads with "Contacted" in Stage\_History) / Total Leads

## **CRM Schema Design**

### **1. Lead Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Lead\_ID (PK) | String | Unique ID |
| Lead\_Name | String | Optional but useful |
| Source | Enum | Web, Referral, Ad, etc. |
| Region\_ID (FK) | Integer | Linked to Region |
| Rep\_ID (FK) | Integer | Assigned sales rep |
| Created\_Date | Date | When lead was created |
| Current\_Stage  Email  Phone  Priority  Last\_Updated | Enum  Varchar  Varchar  Enum  DateTime | Current funnel stage  Lead email to contact  Lead contact  Helps with lead scoring or follow-up logic  The most recent time the lead record was modified or interacted with |
| Stage Sequence | Integer | Numeric order of current stage (e.g., 1 for Captured, 2 for Qualified) |
| Deal\_Source | Enum | Inbound/Outbound |
|  |  |  |
| Company\_ID(FK) | String | Associated company |
| Engagement\_Level | Enum | Hot, Warm, Cold |
| Follow\_Up\_Threshold | Enum | 2 Days, 5 Days, 7 Days, 10 Days |
| Project\_Term | Enum | 0–6 Small, 6–12 Medium\_Small, 12–24 Medium, >24 Large scale |
| Budget\_Tier | Enum | See Deal Table for detailed breakdown |
| Country | String | Country of the lead's company or location |

### **2. Sales\_Rep Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Rep\_ID (PK) | Integer | Primary Key |
| Rep\_Name | String | Sales rep name |
| Email  Phone | Varchar  Varchar | Contact  Rep contact |
| Region\_ID (FK)  Join\_Date  Current\_Status  Performance\_Rating | Integer  Date  Enum  Float | Region assigned  When the rep joined  Status of rep (e.g., Active, Inactive, On Leave)  Out of 0 - 5, updated from KPI metrics |
| Total\_Deals\_Closed | Integer | Number of deals successfully closed by the rep |
| Avg\_conversion\_time | Float | Average time (in days) to convert lead to deal per rep |
| Win\_rate | Float | % of total deals closed (Won / Total Deals Handled) |

### **3. Region Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Region\_ID (PK) | Integer | Primary Key |
| Region\_Name | String | E.g., North, South |
| Region\_Head | String | Name of the manager/head |
| Timezone | String | Timezone of region (e.g., IST) |
| Region\_Target | Float | Revenue target assigned |
| Region\_Status | Enum | Active, Inactive |

Country String Country to which the region belongs

### **4. Stage\_History Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| History\_ID (PK) | Integer | Auto-increment ID |
| Lead\_ID (FK) | String | Link to lead |
| Stage | Enum | Funnel stage (e.g., Lead Captured, Contacted, Demo, Deal) |
| Timestamp  Rep\_ID (FK) | DateTime  Integer | When this stage was reached  Assigned sales rep |
| Stage\_sequence | Integer | Numeric sequence of this stage for chronological tracking |
| Is\_Rentry | Boolean | Indicates if the lead re-entered this stage (true/false) |
| Notes | Text | Optional comments or details from the sales rep at that stage |
|  |  |  |

This table is **critical** for:

* Calculating time spent per stage
* Identifying drop-offs
* Funnel visualizations

### **5. Deal Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Deal\_ID (PK) | String | Unique deal ID |
| Lead\_ID (FK) | String | Related lead |
| Status | Enum | Final result (e.g., Won, Lost, On Hold) |
| Amount  Created\_Date | Decimal  Date | Revenue potential  When deal was created |
| Closed\_Date  Rep\_ID (FK)  Reason\_Lost  Product/Service\_Type (optional) | Date  Integer  Enum  String | When deal was closed  Assigned sales rep  Why deals fail (e.g., Budget, Timing, Competitor)  Segment or offering type (e.g., SaaS, Consulting) |
| Close\_Velocity | Float | Days between Created\_Date and Closed\_Date (calculated field) |
| Deal\_Source | Enum | Deal origin (Inbound, Outbound, Referral, etc.) |
| Feedback\_Notes | Text | Rep's notes or customer feedback on the deal outcome |
| Project\_Type | Enum | Staffing, Fix-bid, Time and Material, Digital Consulting |
| Service\_Category | Enum | Data, Marketing, Cloud, Design, etc. |
| Budget\_Tier | Enum | Short-Term Small, Short-Term Large, Mid-Term Moderate, Mid-Term High, Long-Term Pipeline, Strategic Long-Term |
| Revenue\_Gained | Decimal | Final revenue earned if Status = "Won" |
| Revenue\_Lost | Decimal | Potential revenue lost if Status = "Lost" |

### **6. Company Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Company\_Name | String | Company Name |
| Company\_Industry | Enum | Technology, Healthcare, Finance, Manufacturing, etc. |
| Company\_Size | Enum | Start-up, Small business, Medium Enterprise, Large Enterprise |
| Company\_Ownership | Enum | Public Sector, Private Sector, Joint Venture, Co-op, Franchise |
| Company\_Funding | Enum | Bootstrapped, Seed, Series A/B/C, IPO |
| Country | String | Headquarter country |
| Interest\_Area | Enum | Their area of need (optional) |

## 

## **CRM System Plan**

To simulate a functional CRM environment that supports both historical and real-time behavior, we are implementing a hybrid system using Python, Google Sheets, and Google Apps Script.

**Components Used:**

* **Python**Used to generate realistic mock CRM data using built-in modules (random, uuid, datetime, etc.). Simulates historical data for leads across various stages and sources.
* **Google Sheets**Serves as the primary storage and tracking layer for CRM data. Acts as a centralized database that also powers dashboards and KPIs via formulas and pivot tables.
* **Google Apps Script**Enables automation of data entry, simulates real-time updates, and supports future reporting or alert automation (e.g., trigger-based follow-up reminders).

**CRM Source Strategy:**

We are adopting a **hybrid ingestion approach** to replicate both **past and ongoing CRM activities**:

* **Mock CSV (Historical CRM Data):**
  + Data generated via Python script
  + Covers multiple funnel stages (e.g., Lead → Demo → Deal)
  + Includes multiple records with attributes like source, region, rep, timestamp
* **Google Form (Live CRM Simulation):**
  + Represents new lead entries from marketing, sales, or website forms
  + Connected to the same Google Sheet via Apps Script
  + Auto-triggers Apps Script to assign rep, log funnel stage, and timestamp

## **Data Ingestion Plan**

**Method Chosen:**

**Primary Method (Week 1):**

**Google Sheets + Google Apps Script**

* We are using **Google Sheets** as our CRM simulation platform and **Google Apps Script** to automate the ingestion of new leads, updates to funnel stages, and basic reporting logic.This method enables rapid prototyping and aligns well with our cloud-based workflow.

**Planned for Future (Week 2/3+):**

**Google Sheets API**

* Once the prototype is stable, we will explore using **Google Sheets API**.
* This will allow for more scalable, real-time data handling and production readiness.

**Why This Method?**

We chose Google Sheets combined with Google Apps Script as our initial ingestion method for several reasons aligned with the goals of rapid prototyping, ease of automation, and real-time simulation. Below are the detailed justifications:

* **Ease of Use**Google Sheets and Apps Script are easy to set up and use, with no need for complex infrastructure or local environments.
* **Simulation-Friendly**Supports both CSV uploads and Google Forms, allowing us to simulate historical and real-time lead flows seamlessly.
* **Automation Capable**Apps Script enables scheduled triggers, making it simple to simulate lead creation every few minutes automatically.
* **Integrated with Dashboard Tools**Google Sheets connects directly to tools like Looker Studio, enabling live dashboards without additional setup.
* **Scalable for Early Prototypes**Easily handles thousands of records, which is ideal for early-stage prototypes without performance issues.
* **Free and Cloud-Based**Entirely web-based and free to use, making it ideal for startups or teams with limited resources during early phases.

**Apps Script Prototype: Real-Time CRM Simulation**

**Script Purpose:**

The Google Apps Script serves as a lightweight ingestion engine that automates real-time or scheduled addition of lead records into the CRM system (Google Sheets). It supports simulation of both live entries (via Google Form) and queued historical data (via CSV).

**What the Script Does:**

* **Reads lead data** from either:
  + A connected **Google Form** (real-time user entry), or
  + A **mock data sheet/queue** populated from a CSV file or Python script
* **Generates unique Lead IDs** and timestamps to mimic real CRM behavior
* **Assigns lead attributes** dynamically (e.g., region, rep, source, priority, engagement level)
* **Appends each new lead** into the **Lead** tab in the Google Sheet
* **Inserts an initial funnel stage** entry ("Lead Captured") into the **Stage\_History** sheet, logging the rep, stage sequence, and timestamp
* **Updates metadata** such as the Last\_Updated field and Stage\_Sequence for proper funnel tracking
* **Simulates follow-up stages** using time-based logic or scripted transitions (e.g., from "Lead Captured" to "Contacted" after a delay)

**Execution Modes:**

* **On Form Submit Trigger** – Adds lead immediately upon Google Form submission
* **Time-Driven Trigger** – Based on the timeline given it periodically adds leads for real-time simulation

## **Week 1 Summary: Planning, Setup & Data Source Integration**

In Week 1, we successfully laid the foundation for the CRM Sales Funnel Analytics project. The team defined a clear project scope focused on optimizing the lead-to-deal conversion pipeline through data-driven analysis. We identified funnel stages, assigned team roles, created mock datasets, and implemented an automated data ingestion strategy using Google Sheets and Apps Script.

We designed a hybrid CRM simulation system that captures both historical and live CRM data. A Python-based generator was used to simulate past leads across funnel stages, while Google Forms with Apps Script support real-time lead entry. A dynamic CRM schema was finalized, and the ingestion mechanism was prototyped to populate and track leads with stage history and attributes like source, region, and assigned rep.

This foundation will support KPI calculation, conversion tracking, funnel performance monitoring, and regional/rep-level insights in future weeks.

**Week 1 Deliverables Checklist**

**1. Funnel Stage Definitions** Defined and documented key funnel stages: *Lead Captured → Qualified → Contacted → Demo → Proposal → Negotiation → Deal Won/Lost*, including optional stages like *Responded* and *Demo Completed* for deeper tracking.

**2. Project Plan (Roles, Tools, Timeline)** Team roles for week 1 were assigned clearly:

* *Anurag:* Project scope and funnel logic
* *Aman:* KPI definitions and reporting
* *Chandini:* Schema design and Google Sheets setup
* *Hamsapriya:* Mock data generation
* *Bala Ramana:* Ingestion script and documentation

**Tools selected:** Google Sheets, Python, Apps Script, Google Forms.  
**Timeline:** Week-wise milestones outlined starting from planning to dashboarding.

**3. Initial CRM Data Files (Mock/Simulated)**Generated realistic mock lead data using Python (random, uuid, datetime) simulating historical leads and funnel movement. CSVs cover tables like Lead, Deal, Stage\_History, Sales\_Rep, etc.

**4. Data Ingestion Script Prototype**Script design is documented and scoped:

* Will ingest new leads from Google Form
* Will generate Lead\_IDs, assign reps, and update Stage\_History
* Trigger modes (form-based and time-based) are planned

**5. Schema Design Document** Finalized schema includes six structured tables:

* Lead, Deal, Stage\_History, Sales\_Rep, Region, Company  
  Each table includes key fields, datatypes, relationships, and additional metadata required for tracking funnel movement, lead sources, performance indicators, and more.