HCL MaxAI 25.1.0 Guidelines Document

1. Introduction to the System

The system is an advanced AI-powered platform designed to streamline **data analysis**, **documentation search**, and **report interpretation** within the Unica product ecosystem. It integrates natural language processing (NLP), document embeddings, and advanced LLM capabilities to deliver precise and actionable insights.

Purpose

The system assists users by:

- **Answering Analytics Questions**: Queries related to campaigns, journeys, deliverables, or other metrics.
- **Document Search**: Fetching and summarizing relevant content from Unica documentation.
- **Report Explanation**: Parsing JSON-based reports to provide clear interpretations.

Key Features

1. **Context-Aware Querying**:
    - Handles multi-step interactions to maintain conversational context.
    - Rephrases vague questions into specific, actionable queries.
        i. **This has been enabled to enhance "Usability" of the system but it can lead to hallucination if incoming questions are not properly formatted. 'Best Practices Below'.**
2. **Embedded Knowledge**:
    - Supports Unica product knowledge (e.g., Campaign, Journey, Deliver, etc.).
    - Uses embeddings for fast and accurate document search.
3. **SQL Generation**:
    - Generates SQL queries for complex analytical questions.
    - Follows specific table mappings and data model guidelines.
4. **Dynamic Error Handling**:
    - Detects ambiguous or unsupported queries and asks for clarification.

Capabilities

- **Multi-Domain Support**: Answers queries related to:

    - Campaign metrics (e.g., response rates, revenue).
    - Journey analysis (e.g., drop-off rates, milestone progress).
    - Deliverables (e.g., email, SMS performance).
- **Document Summarization**:
    - Extracts relevant sections of Unica documentation.
    - Simplifies technical jargon into user-friendly summaries.
- **Report Parsing**:
    - Processes structured report data (e.g., JSON) to generate insights.
    - Answers user questions based on provided reports.

Critical Design Consideration

The system employs a **single conversation interface** for all three capabilities—**document search**, **report analysis**, and **analytics queries**.

- This unified design can lead to **context switching challenges**, as the system processes all interactions via a **classification engine**.
- To minimize hallucinations caused by switching between capabilities:
    - The system only retains the **last two conversations** for context.
    - This design deliberately limits deeper conversational history to avoid random outputs or irrelevant answers.

System Evolution and Non-Deterministic Behavior:

- As the system evolves, it will learn from:
    - User interactions
    - Success and failure patterns
    - Feedback loops
- This iterative process means **prompts and responses will improve over time**, but users should not expect high accuracy or perfect responses on Day 0.
- **Non-Deterministic Nature**:
    - LLMs (Large Language Models) inherently produce **non-deterministic outputs**, meaning responses may vary slightly for the same question across different users or sessions.
    - **Analytical Query Note**:
        - While the **values in responses** should always be accurate, the **presentation** (e.g., format, phrasing) might vary depending on how context was retained or interpreted.

Important: This approach is not a limitation but a conscious trade-off to balance **context accuracy** and **response reliability**. Users must:

- Reset the context when switching between unrelated queries.
  - Not currently Available but being managed via limited depth of the conversation.
- **Be mindful of the context carried forward within a session.**

Points to be mindful of while using the system

1. **Ambiguity Sensitivity**:
   - Vague or incomplete queries may result in clarification prompts.
   - Example: "What’s the performance of Campaign X?" (What metric: audience, revenue, or conversion?)
2. **Context Dependency**:
   - Cross-context questions (unrelated queries in one session) may cause errors.
   - Use the **Reset Context** feature to avoid confusion.
3. **Data Dependency**:
   - The system relies on accurate and updated data models.
   - If underlying data is incomplete, results may be incorrect, even with correct SQL queries.

2. Best Practices for Interacting with the System

Interacting effectively with the system requires understanding its capabilities and constraints. This section provides actionable guidance for crafting precise queries, avoiding common pitfalls, and leveraging the system's strengths.

Understanding the Query Structure

Every good query is built using a clear and logical structure:

1. **Action**: What do you want the system to do?
   Examples: "Show," "List," "Provide," "Explain," "Compare."
2. **Subject**: What is the query about?
   Examples: Campaigns, Journeys, Offers, Audience.
3. **Modifiers**: Add specific details to narrow the scope.
   Examples: Timeframes, specific metrics, filters, comparisons.

Examples:

- Bad: "What's my data?"
- Good: "What is the response rate for Campaign X over the last 30 days?"

Key Guidelines for Query Formation

1. **Be Specific**:
   - **Bad**: "Show me data."
   - **Good**: "Show the open rate for emails in Campaign X in the last week."
2. **Provide Context**:
   - **Bad**: "Give me results for the same campaigns."
   - **Good**: "Give me results for Campaign A and Campaign B from the last query."
3. **Stick to Supported Topics**:
   - Only ask questions related to supported domains (e.g., campaigns, journeys, deliverables).
   - **Example**: "What is the response rate for Offer Y in Campaign Z?"
4. **Follow the Data Model**:
   - Use appropriate table prefixes for clarity:
     - **Campaigns**: Use `UARC`.
     - **Delivery**: Use `UARE`.
5. **Avoid Ambiguity**:
   - **Bad**: "What's the performance of my campaign?"
   - **Good**: "What is the audience engagement rate for Campaign Y?"

6. **Avoid Unsupported Scenarios**:
    - Don’t ask questions beyond the system's capabilities.
    - **Bad**: "What is the sentiment of my campaign emails?"
    - **Good**: "What is the open rate for my campaign emails?"

Resetting Context

The system operates with a limited conversational memory (last two queries) to minimize hallucinations. Users must be mindful of:

- Avoiding context switching mid-conversation.
- Resetting context when starting a new topic (feature not yet available in Beta).

Example:

- **Bad**: "What’s my audience size?" followed by "Show me response rates for Journey X."
- **Good**: "What is the audience size for Journey X?" followed by "What is the response rate for Journey X?"

Semantic Framework for Query Formation

Treat queries as structured sentences with defined roles:

1. **Who/What**: Identify the subject.
    - Example: "Campaign X," "Email channel."
2. **What Metric**: Define the desired outcome.
    - Example: "Response rate," "Audience size."
3. **When**: Add a timeframe or condition.
    - Example: "In the last 30 days," "Compared to last month."
4. **Why/How**: Optional clarification.
    - Example: "Why has the response rate dropped?" or "How does it compare to Campaign Y?"

Examples:

- **Simple Query**: "Show the response rate for Campaign X."
- **Complex Query**: "Compare the click-through rates of emails and WhatsApp messages for Campaign Z over the last 30 days."

Examples of Good, Bad, and Ugly Queries

Query	Classification	Why
"What are campaigns running today?"	Good	Clear action and subject with a time constraint.
"What are channels being used in them?"	Good	Logical follow-up query.
"How many contacts I made today?"	Good	Specific and scoped.
"Tell me data."	Bad	Ambiguous; no clear subject or metric.
"What's the performance of Campaign X?"	Bad	Vague; performance needs clarification (e.g., audience, revenue, response rate).
"Send me count of emails send daily? and what is Click through rate?"	Ugly	Mixed queries; should be broken into two separate questions.
"How can I create segments in Campaign?"	Good	Clear and actionable.
"Is Email better or WhatsApp? Any suggestions?"	Ugly	Unsupported subjective comparison query.
"Show me performance of milestones for Journey X."	Good	Focused and measurable query.
"I am lost in my campaign. What went wrong?"	Ugly	Overly conversational; lacks actionable structure.
"What is the best time to send an email campaign?"	Good	Specific and measurable.
"How are Deliver Campaign and Mailings different?"	Good	Structured comparison request.

User-Defined Guardrails for Input Queries

Empowering users to set their own boundaries within queries helps maintain precision. Examples:

1. **Set Time Constraints**:
   - "Show me data only for the last 7 days."
   - "Exclude results older than 30 days."
2. **Define Metrics Clearly**:
   - "Only show response rates and unique responders."
3. **Exclude Certain Data**:
   - "Ignore data from campaigns marked as 'archived.'"

4. **Combine Filters**:
    - "Show campaigns using email channels with a response rate above 20% in the last month."

Illustrative Example:

- **Bad**: "How is my campaign doing?"
- **Good with Guardrails**: "Show me the response rate, unique responders, and audience size for Campaign X using email channels in the last 30 days."

Advanced Query Examples

1. **Cross-Metric Comparison**:
    - "What is the response rate for Offer A compared to Offer B in Campaign X over the last month?"
2. **Trend Analysis**:
    - "Show the week-over-week performance of Campaign Y in terms of audience engagement."
3. **Performance Breakdown**:
    - "What is the milestone progress and drop-off rate for Journey Z?"
4. **Actionable Insights**:
    - "Based on past performance, which offer should I use for Campaign X?"

Bad Queries

3. **Generic Input**:
    a. "Show campaigns."
    b. Problem: Generates SQL without filters, e.g., `SELECT  FROM campaigns`.
4. **Ambiguous Nouns**:
    a. "Show active data."
    b. Problem: SQL cannot determine what "data" refers to.
5. **Unclear Context**:
    a. "List journeys from last week."
    b. Problem: SQL might interpret "last week" incorrectly without specific dates.

Good Queries

6. **Specific Attributes**:
    a. Input: "Show campaigns created in January 2024."
    b. SQL: `SELECT  FROM campaigns WHERE created_date BETWEEN '2024-01-01' AND '2024-01-31'`.
7. **Explicit Filters**:
    a. Input: "Show emails sent for Campaign X."

   b. SQL: `SELECT email_subject, sent_date FROM email_logs WHERE campaign_id = 'X'`.

8. **Clear Scope**:
   a. Input: "Get journeys with over 10% conversion rate."
   b. SQL: `SELECT name, conversion_rate FROM journeys WHERE conversion_rate > 10`.

Enhancing User Training for SQL Generation: Addressing "Where Clause" Issues (NOUN Should be highlighted as part of question and it should be clear. Guardrail being created for prompting user zs well)

When generating SQL queries from user inputs, **ambiguous or vague use of nouns** (e.g., "campaign," "journey") in the query can lead to poorly constructed `WHERE` clauses. This issue arises because the system interprets the nouns literally, potentially resulting in overly broad or incorrect SQL outputs.

Question formatting Guidelines

---

1. Simple Analytical Query

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Modifiers]**

Example:

- "Show [Action] the response rate [Predicate] for Campaign A [Subject] over the past week [Modifiers]."

---

2. Multi-Part Query

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Modifiers]** + **[Temporal Indicator]** + **[Comparison/Qualifier]**

Example:

- "List [Action] all journeys [Subject] that had more than 50% completion rates [Predicate] last quarter [Temporal Indicator] and compare them to the current quarter [Comparison]."

---

3. Contextual Query with Metadata

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Modifiers]** + **[Metadata/Contextual Clause]**

Example:

- "Analyze [Action] the milestones [Subject] achieved in Journey X [Predicate] and show their completion dates [Modifiers] using the UARC database [Metadata]."

---

4. Detailed Temporal and Logical Query

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Temporal Indicator]** + **[Logical Operators]**

Example:

- "Show [Action] all campaigns [Subject] that were active last month [Temporal Indicator] AND had an ROI above 20% [Logical Operators]."

---

5. Comparative Query

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Comparison/Qualifier]**

Example:

- "Compare [Action] the response rates [Subject] for Campaign A and Campaign B [Comparison] to identify which performed better [Predicate]."

---

6. Multi-Subject Query

Structure:
[Action] + **[Multiple Subjects]** + **[Predicate]** + **[Modifiers]**

Example:

- "List [Action] the milestones and goals [Multiple Subjects] for Journey Y [Predicate] and show their status [Modifiers]."

---

7. Directive-Based Query

Structure:
[Action] + **[Subject]** + **[Directive]** + **[Temporal Indicator]**

Example:

- "Summarize [Action] the key performance indicators [Subject] for all journeys [Directive] over the past 6 months [Temporal Indicator]."

---

8. Hypothetical Analysis Query

Structure:
[Action] + **[Subject]** + **[Hypothetical Predicate]** + **[Modifiers]**

Example:

- "If [Hypothetical Predicate] 200 more contacts are sent to Campaign X [Subject], what will the response rate be [Action]?"

---

9. Operational Query

Structure:
[Action] + **[Subject]** + **[Procedural Predicate]**

Example:

- "Explain [Action] how to create a new segment [Subject] in Unica Campaign [Procedural Predicate]."

---

10. Query with Contextual and Temporal Constraints

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Temporal Indicator]** + **[Contextual Clause]**

Example:

- "Provide [Action] a summary of Campaign Z's performance [Predicate] over the last quarter [Temporal Indicator], considering its use of email and SMS channels [Contextual Clause]."

---

11. Nested Query

Structure:
[Action] + **[Subject]** + **[Nested Predicate]** + **[Temporal Indicator]**

Example:

- "Show [Action] all journeys [Subject] where email was the primary channel [Nested Predicate] and compare their performance over the last two months [Temporal Indicator]."

12. User-Defined Guardrails Example

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[User-Specific Rules]**

Example:

- "List [Action] all touchpoints [Subject] in Journey A [Predicate], ensuring only completed milestones are included [User-Specific Rules]."

13. Summary and Suggestion Query

Structure:
[Action] + **[Subject]** + **[Predicate]** + **[Suggestions/Predictions]**

Example:

- "Summarize [Action] the past week's performance [Subject] for all email campaigns [Predicate] and suggest the best time to send future emails [Suggestions/Predictions]."

Section 3: What to Avoid

1. Ambiguity: Avoid Vague or Unclear Queries

Ambiguity in queries makes it difficult for the system to understand the user's intent, leading to incomplete or incorrect responses. A well-defined question eliminates uncertainty and ensures the system provides accurate outputs.

- Examples:

  - Bad: "What’s the performance of my campaign?"

    - Why It’s Bad: "Performance" is vague and could refer to audience engagement, response rates, or revenue.

  - Good: "What is the response rate for Campaign A over the past week?"

    - Why It’s Good: Clearly specifies the metric (response rate), campaign (Campaign A), and timeframe (past week).

- Additional Examples:

  - Bad: "Show journey details."

    - Why It’s Bad: What details? Status? Milestones? Touchpoints?

  - Good: "List all active journeys and their milestones."

    - Why It’s Good: Specifies the scope (active journeys) and the type of information (milestones).

---

2. Unsupported Scenarios: Stay Within the System’s Scope

The system is designed to handle specific domains (e.g., campaigns, journeys, reports). Queries outside this scope may result in irrelevant or nonsensical answers.

- Examples:

  - Bad: "Tell me the sentiment of campaign emails."

    - Why It’s Bad: The system doesn’t support sentiment analysis.

  - Good: "Show the open rate and click-through rate for Campaign A emails."

    - Why It’s Good: Focuses on measurable email performance metrics.

- Additional Examples:

  - Bad: "How do I make green tea?"
    - Why It’s Bad: Irrelevant to the system’s capabilities.
  - Good: "What are the response rates for email campaigns sent this week?"
    - Why It’s Good: Falls within the system’s analytics capabilities.

---

3. Cross-Context Errors: Avoid Unrelated Queries in the Same Session

When users mix unrelated questions in the same session, the system’s context retention can lead to errors. It may merge or confuse topics, producing hallucinated or inaccurate responses.

- Examples:
  - Bad:
    - Query 1: "What’s the response rate for Campaign A?"
    - Query 2: "Show document details for Journey B."
    - Why It’s Bad: Mixing analytics and document search confuses the system.
  - Good:
    - Query 1: "What’s the response rate for Campaign A?"
    - Reset Context: "Switch to document search."
    - Query 2: "Show document details for Journey B."
    - Why It’s Good: Clear transition between unrelated queries.

- Additional Examples:
  - Bad: "List campaigns. How do I install Unica?"
    - Why It’s Bad: Combines analytics and installation instructions.

- Good:
  - Query 1: "List all campaigns running this month."
  - Reset Context: "Switch to Unica installation guide."
  - Query 2: "Show steps to install Unica Campaign."

Tips to Avoid These Pitfalls

1. Be Specific:
   - Clearly state what you want and include all relevant details.
   - Example: "Show the click-through rate for Campaign A emails sent last week."

2. Stay on Topic:
   - Group queries by type (analytics, reports, or document search) within a session.
   - Example:
     - Query 1: "Show all active journeys."
     - Query 2: "List touchpoints for Journey B."
     - Avoid adding: "What are Unica installation steps?"

3. Use Reset Context:
   - When switching to a different domain or topic, indicate a transition.
   - Example: "Reset context. Show Unica document details for campaign workflows."

4. Understand System Scope:

- Stick to supported functionalities like campaign performance, journey analysis, or documentation queries.

- Avoid unsupported features like real-time sentiment analysis or unrelated general queries.

5. Review Query Structure:

- Use the Action-Subject-Modifier approach for clarity:

  - Action: What you want (e.g., Show, List, Provide).

  - Subject: The focus of the query (e.g., Campaigns, Journeys, Offers).

  - Modifier: Additional details (e.g., timeframes, metrics, filters).

Section 5: How the System Responds to Errors

This section explains how the system handles scenarios where it cannot generate a response due to incomplete or unclear inputs. It provides users with insights into fallback mechanisms, error messaging, and clarification requests.

System Error Response Mechanisms

1. Fallback Prompts

When the system encounters an error or lacks sufficient information to generate a response, it triggers fallback prompts to guide the user:

- **Example:**
  - **Input:** "What is the performance of my campaign?"
  - **Fallback Prompt:** "Could you specify what aspect of performance you are looking for? For example, audience engagement, revenue, or response rates?"
- **Why It Happens:** The query is too broad, and the system needs more details to provide an accurate response.

---

2. Clarification Requests

The system may ask for additional input to refine the query:

- **Example:**
  - **Input:** "Show me campaigns for this week."
  - **Clarification Request:** "Are you asking for active campaigns, completed campaigns, or campaigns by specific metrics?"
- **Why It Happens:** The query lacks specificity, leading to potential ambiguity in the system's response.

---

3. Default Error Messages

When the system cannot find relevant data, it provides polite error messages:

- **Example:**
  - **Input:** "Tell me the sentiment of campaign emails."
  - **Error Message:** "Sorry, I cannot analyze sentiment as it is outside the scope of this system. Please ask about supported analytics or documentation."
- **Why It Happens:** The query is beyond the system's capabilities or unrelated to its design.

---

4. Handling Context-Related Errors

If the system identifies cross-context issues (e.g., switching between reports and analytics without resetting), it will attempt to guide the user back on track:

- **Example:**
  - **Input:** "What is the click-through rate of Campaign X? Also, explain how Unica handles email delivery."
  - **Response:** "Your first question pertains to analytics, and the second to documentation. Please clarify if you want the system to prioritize analytics or documentation."

User Feedback Mechanisms

1. Explanation of Error Causes

The system may explain why it couldn’t generate a response:

- **Example:**
    - **Input:** "What are the best practices for journeys?"
    - **Response:** "The input seems related to Unica documentation. If that’s correct, could you rephrase it to include a product name or specific feature?"

2. Error Highlighting

The system highlights problematic parts of the input:

- **Example:**
    - **Input:** "Tell me about campaigns and offers."
    - **Response:** "The query is too broad. Could you specify if you need metrics for campaigns or offers?"

System Safeguards

1. Guardrails for Noun Overloading

- When the system detects overly generic nouns (e.g., "campaigns," "offers") without modifiers, it requests specificity.
    - **Example:**
        - **Input:** "Show me campaigns."
        - **Response:** "Are you asking about active campaigns, campaigns by status, or campaigns by performance?"

2. Prevention of Cross-Context Hallucination

- The system limits conversational history to reduce context misinterpretation.
    - **Example:**
        - **Input:** "How many emails were sent today?"
        - **Follow-Up:** "What is sentiment analysis?"
        - **Response:** "Sentiment analysis is unrelated to the previous query. Could you clarify your intent?"

What Users Should Do

1. Rephrase the Query

- If the system asks for clarification, reframe the question with more details.
    - **Bad:** "How is my campaign doing?"

    - **Good:** "What is the response rate and revenue for Campaign X?"

2. Reset Context

- If switching between analytics, documents, and reports, reset the context for better results.
    - **Example:** "Reset and analyze the performance of Journey A."

3. Stay Within Scope

- Avoid asking questions outside the system's design.
    - **Bad:** "What's the weather today?"
    - **Good:** "What is the conversion rate for Journey B?"

Examples of Error Scenarios

Ambiguity

- **Input:** "Show performance."
- **Response:** "Could you specify what performance you are referring to? Campaigns, journeys, or offers?"

Unsupported Queries

- **Input:** "Tell me the mood of email responses."
- **Response:** "The system does not analyze mood or sentiment. Please ask about metrics or performance."

Cross-Context Mixing

- **Input:** "Explain how journeys work and show me campaign results."
- **Response:** "These queries belong to different contexts. Please reset or clarify your query."

Section 6: FAQs

This section addresses common questions and concerns users might have while interacting with the system. It provides clear explanations and actionable advice to ensure smooth and effective usage.

---

1. Why does the system ask me to clarify?

Explanation:

The system prompts for clarification when it identifies ambiguity or insufficient context in the query. This ensures that the response aligns closely with the user's intent.

Examples:

- **Input:** "Show me performance."
  - **Clarification Prompt:** "Are you asking about campaign performance, journey performance, or something else?"
- **Input:** "What is the response rate?"
  - **Clarification Prompt:** "Could you specify which campaign or journey you are referring to?"

What You Can Do:

- Be specific in your queries by including relevant details like campaign names, timeframes, or metrics.
  - **Bad:** "Show me performance."
  - **Good:** "Show me the email open rate for Campaign X in the last month."

---

2. What should I do if the system doesn't understand my query?

Explanation:

If the system fails to interpret the query, it may:

1. Return a fallback prompt requesting clarification.
2. Highlight parts of the query it couldn't process.
3. Inform you that the requested functionality is outside its scope.

Examples:

- **Input:** "Tell me the best time for emails."

  - **System Response:** "The system does not provide recommendations on email timings. Please ask about email performance metrics."
- **Input:** "What's the sentiment of Campaign X?"

  - **System Response:** "The system cannot analyze sentiment. Please ask about supported analytics or documentation."

What You Can Do:

- Rephrase your query to be more specific.
  - **Bad:** "Show me everything."
  - **Good:** "Show me the response rate and audience engagement for Campaign X."
- Use terms aligned with the system's capabilities and data model (e.g., UARC for campaigns, UARE for delivery).

---

3. How does the system decide which context to use?

Explanation:

The system uses a classification engine to determine whether the query pertains to documents, reports, or analytics. However, overlapping queries or ambiguous phrasing may lead to misclassification.

Examples:

- **Input:** "What is the click-through rate of Campaign X? Also, explain email delivery."
  - **System Response:** "Your query spans multiple contexts (analytics and documentation). Please clarify which part you want prioritized."

What You Can Do:

- Avoid combining multiple contexts in a single query.
  - **Bad:** "What is the performance of Campaign X, and how does email delivery work?"
  - **Good:** "What is the performance of Campaign X?" (analytics context)
  - Follow-Up: "Explain email delivery." (documentation context)

---

4. What if I make a typo or grammatical error?

Explanation:

The system attempts to interpret queries despite minor typos or grammatical errors. However, significant errors may lead to misinterpretation or irrelevant results.

Examples:

- **Input:** "Show campigns peformence."
  - **System Response:** "Are you asking about campaign performance? Please confirm or rephrase."

- **Input:** "How maany audienss entered Journey X?"
  - **System Response:** "Did you mean 'How many audiences entered Journey X?' Please clarify."

What You Can Do:

- Double-check your query for accuracy before submitting.
- If prompted, confirm the system’s interpretation or rephrase your query.

---

5. Can the system answer follow-up questions?

Explanation:

Yes, the system retains limited conversational history (last two interactions) to support follow-up questions. However, this trade-off is designed to reduce context-switching errors.

Examples:

- **First Query:** "Show me Campaign X performance."
- **Follow-Up Query:** "What is the response rate?"
  - **System Response:** "The response rate for Campaign X is 15%."

What You Can Do:

- Ensure follow-up questions are directly tied to the immediate prior query.
  - **Bad:** "What is sentiment analysis?"
  - **Good:** "What is the click-through rate for Campaign X from the previous results?"

---

6. What happens if my query is out of scope?

Explanation:

If the query asks for unsupported functionality (e.g., sentiment analysis or unrelated topics), the system will:

1. Inform you that the query is out of scope.
2. Suggest alternative supported queries.

Examples:

- **Input:** "Tell me the sentiment of my emails."

    - **System Response:** "Sentiment analysis is not supported. You can ask about email open rates or click-through rates instead."
- **Input:** "What is the weather today?"

    - **System Response:** "Sorry, I cannot answer unrelated queries. Please ask about campaigns, journeys, or offers."

What You Can Do:

- Familiarize yourself with the system's capabilities using the "Supported Queries" section of this guide.
- Reframe your query to align with supported use cases.

---

7. How can I provide feedback on system responses?

Explanation:

User feedback is essential for improving the system's performance. If a response seems incorrect, incomplete, or unclear:

1. Use the feedback button to report issues.
2. Specify what was wrong with the response (e.g., incorrect data, ambiguous phrasing).

What You Can Do:

- Provide constructive feedback:
    - **Example Feedback:** "The system provided revenue data instead of response rate for Campaign X."
- Suggest improvements:
    - **Example Feedback:** "It would be helpful if the system could explain how click-through rate is calculated."

Segmentation Agent User Guide

PURPOSE

This guide helps marketers—without deep SQL or schema knowledge—to write clear, precise natural language prompts for our AIpowered segmentation engine. Follow the patterns, examples, and templates below to express segmentation criteria from simple to highly complex scenarios.

CONTENTS

1. PROMPT COMPONENTS & BEST PRACTICES

- Use clear, unambiguous language (“age over 30”, “state is California”).
- State filters first, then aggregations (“who made purchases in last 90 days and whose total spend exceeds $5,000”).
- Number multistep criteria when listing more than two conditions.
- Specify time windows (“last 30 days”, “Q3 2023”).
- Combine AND/OR carefully: describe OR groups together (“A or B”), and AND lists item by item.
- For subgroups, describe the main group then the subgroup requirement.
- If you need comparative or percentile logic, name it (“top 10% of spenders”).

2. PROMPT TEMPLATES & EXAMPLES

A. SIMPLE FILTERS

“Show me customers who [FIELD] is [VALUE].”

Examples:

- Show me customers whose gender is Female.
- Show me customers who state is California.

B. TWOCONDITION AND

“Find customers who [COND1] and [COND2].”

Examples:

- Find customers who age is over 30 and income level is High.
- Find customers who city is New York and made a purchase in the last 30 days.

C. TWOCONDITION OR

“Give me customers who [COND A] or [COND B].”

Examples:

- Give me customers who responded to campaign X or have spent more than $5,000.
- Give me customers whose education level is PhD or Master’s Degree.

D. TIMEBOUND + AGGREGATE

“Identify customers who [ACTION] in the last [TIMEFRAME] and whose [METRIC] is [THRESHOLD].”

Examples:

- Identify customers who made purchases in the last 90 days and whose total spend is over $10,000.
- Identify customers who opened an account in the last month and whose number of transactions ≥ 5.

E. SUBGROUP CONDITIONS

“Select customers who [PRIMARY FILTER], and among them those who [SECONDARY FILTER].”

Examples:

- Select customers who responded to campaign ABC123, and among them those whose spending places them in the top 10%.
- Select customers who live in Texas, and among them those who made more than 3 purchases last quarter.

F. EXCLUSIONS

“Find customers who [COND] but exclude those who [EXCLUSION].”

Examples:

- Find customers who use our mobile app but exclude those who have closed their credit card.
- Find customers with txn_amount > 1,000 but exclude those in the Low income bracket.

G. MULTISTEP PARAGRAPH

“I need customers who meet all of these:

1. [Filter A]
2. [Filter B]
3. [Filter C]
4. [Filter D]

And then within that set, find those who [Subgroup condition].”

Example:

I need customers who:

1. live in California,
2. responded to our Spring campaign via email,
3. are over 40 years old,
4. have made at least 5 purchases this year.

And then within that set, find those whose total spend ranks in the top 5%.

H. COMPARATIVE SEGMENTS

“Compare [Group A] versus [Group B], based on [Metric].”

Examples:

• Compare customers who are married versus single, based on their average transaction amount in Q2.

• Compare those who clicked Offer A versus Offer B, based on total spend last month.

I. MEDIUM COMPLEXITY (3–5 ATTRIBUTES + AGGREGATES)

1. Find customers aged 25–35 in California with incomelevel High, who made ≥ 10 transactions in the last 60 days and whose average transaction amount > $200.
2. Identify female customers with a Master’s Degree, from New York or New Jersey, who responded to campaign “SUMMER21” and made total purchases > $5,000 in Q3 2023.
3. Show customers whose maritalstatus is Married, lifestylepreferences include “Luxury”, and whose last 5 transactions sum to more than $2,500.

J. HIGH COMPLEXITY (5+ ATTRIBUTES, SUBGROUPS, PERCENTILES)

1. I want customers who:
   a. are Male or Nonbinary,
   b. have incomelevel Very High,
   c. have a PhD or Master’s degree,
   d. responded to campaign “REWARDS2024” within last 120 days via SMS,
   e. and among those, find the top 20% by total txn_amount in the last year.
2. Identify customers who:
   a. live in Texas or Florida,
   b. are over 45 years old,
   c. made at least one purchase with merchant “XYZ_CORP”,
   d. have closed_Products CONTAINS “Auto Loan”,
   e. and whose aggregate txn_amount is in the 75th percentile across all such customers.
3. Select customers who:
   1. have buyingmotivations “Quality” and “Brand”,
   2. used DEVICE_OS “iOS” in last 30 days,
   3. clicked offercode “UPG2023” or “UPG2024”,
   4. have success_flg “UA” in UnicaCH,
   5. and whose median monthly spend over the past 6 months > $1,000.

3. BREAKING DOWN A PROMPT

1. **Identify Datasets or Tables**
   e.g., Transactions, Conversions, Demographics, Behavior.
2. **List Filters**
   Simple equality: field value
   Range: field > X, field BETWEEN A and B
   Date logic: “last X days”
3. **Specify Aggregations**
   SUM(txn_amount), COUNT(transactions), AVG(...)

   Percentile or topN language (“top 10%”)
4. **Define Logical Structure**
   AND vs OR groups
   Nested subgroup: “among those” / “within that set”
5. **Clarify Output**
   Usually “customers who …” implies select distinct customer_id

4. FAQ
Q: What if I don’t know the exact field name?
A: Use logical terms (e.g., “age”, “spend”, “campaign code”)—our system maps them to schema.

Q: How do I express percentile filters?
A: Say “top X% by [metric]” or “in the 75th percentile of [metric]”.

Q: Can I mix AND and OR?
A: Yes—group OR conditions together in a single clause:
  “who A or B, and who C”

Q: What if I need to exclude someone?
A: Use “but exclude those who [condition]”.

Q: How do I request subgroups?
A: State the main group first, then “among them those who …”.

5. BLANK PROMPT TEMPLATES
(Simple → Complex)

1. “Show me customers who [FIELD] is [VALUE].”

2. “Find customers who [COND1] and [COND2].”

3. “Give me customers who [COND A] or [COND B].”

4. “Identify customers who [ACTION] in the last [TIMEFRAME] and whose [METRIC] is [THRESHOLD].”

5. “Select customers who [PRIMARY FILTER], and among them those who [SECONDARY FILTER].”

6. “Find customers who [COND] but exclude those who [EXCLUSION].”

7. “I need customers who meet all of these:
   1. [Filter A]
   2. [Filter B]
   3. [Filter C]

   And then within that set, find those who [Subgroup condition].”

8. “Compare [Group A] versus [Group B], based on [Metric].”

9. “[MEDIUM] Find customers aged [X–Y] in [STATE] with [ATTRIBUTE], who made ≥ [N] transactions in the last [DAYS] and whose average txn_amount > [AMOUNT].”

10. “[HIGH] I want customers who:

    a. [Filter 1],

    b. [Filter 2],

    c. [Filter 3],

    d. responded to [CAMPAIGN] within last [DAYS],

    e. and among those, find the top [PERCENT]% by [METRIC].”