AI Automation / AI Agents

Build an agent that queries databases, generates insights, and creates visualizations automatically.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Business Intelligence, Reporting, Data Democratization
Updated: May 2026
Why This Prompt Exists
Data analysis is bottlenecked by two things: writing SQL queries and interpreting results. An AI agent can do both — but needs the right tools and prompts.

You get:

- agents that generate wrong SQL (insecure or inefficient queries)
- agents that misinterpret results (correlation vs. causation mistakes)
- agents that can’t handle complex multi-table joins
- no visualization (just numbers, no charts)
- no way to ask follow-up questions (one-and-done analysis)

But data agents need structure:

- schema awareness: what tables and fields exist?
- query generation: natural language → SQL
- execution safety: read-only, timeouts, row limits
- insight extraction: what does this result mean?
- visualization: appropriate chart type for the data

Without design, data agents are dangerous.

This prompt designs safe, effective data analysis agents.

The Prompt

```
Assume the role of a data automation architect who designs AI data
analysis agents.

Your task is to create an agent that queries data and generates
insights.

Generate:

1. DATA SOURCES
   - Database(s) available
   - Key tables and fields (schema summary)
   - Update frequency (real-time / daily / weekly)

2. QUERY CAPABILITIES
   - Question types supported: [aggregation / filtering / trend /
comparison / prediction]
   - Question types NOT supported: [e.g., "causation questions"]
   - Join complexity: [simple (2 tables) / moderate (3-5 tables) /
complex]

3. SAFETY RULES
   - Read-only (no INSERT, UPDATE, DELETE)
   - Row limit (e.g., "never return more than 10,000 rows")
   - Timeout (e.g., "queries that run > 30 seconds are cancelled")
   - Sensitive data restrictions (e.g., "don't expose PII")
```

4. QUERY GENERATION PROTOCOL
   - Step 1: Parse natural language question
   - Step 2: Map to available tables/fields
   - Step 3: Generate SQL with comments
   - Step 4: Estimate result size (warn if too large)
   - Step 5: Execute after user approval

5. INSIGHT EXTRACTION
   - For each result: what does this mean for the business?
   - Statistical significance (if applicable)
   - Anomaly detection (what's unexpected?)
   - Comparison to historical baselines

6. VISUALIZATION RULES
   - Time series → line chart
   - Categories comparison → bar chart
   - Part-to-whole → pie/bar chart (prefer bar)
   - Distribution → histogram
   - Correlation → scatter plot

7. READY-TO-USE AGENT PROMPT
   - The system prompt for the data analysis agent

INPUTS:

Database schema (tables and fields):
[PASTE OR DESCRIBE]

Typical questions users ask:

```
[E.G., "How many signups last week?", "What's our retention rate?"]

User technical level:
[NON-TECHNICAL / ANALYST / DATA SCIENTIST]

Data volume:
[SMALL (<1M rows) / MEDIUM (1M-100M) / LARGE (>100M)]

RULES:
- Always use read-only database connections (prevent accidents)
- Set aggressive row limits for exploratory queries
- Pre-validate SQL for syntax errors before execution
- Flag results that exceed statistical or practical significance thresholds
- Log all queries for audit and optimization
- Provide explanations of results in business terms, not just statistics
```

How To Use It

- Always use read-only database connections for AI agents — one wrong update is catastrophic.
- Set aggressive row limits (1000 rows) for exploratory queries.
- Log all queries for audit and optimization.
- Provide business explanations for results, not just statistics.
- Flag results that might be statistically significant but practically meaningless.

Example Input

Database schema:
“Users table (user_id, signup_date, plan_type, country). Payments table (payment_id,

user_id, amount, date).”

Typical questions users ask:
“Monthly revenue, signups by country, retention by plan type”

User technical level:
“NON-TECHNICAL”

Data volume:
“MEDIUM”

Why It Works
Most data agents try to answer any question — which leads to wrong SQL, misinterpreted results, and dangerous queries.

This framework improves outcomes by forcing:

- schema awareness (what data is available?)
- query capability boundaries (what questions can it answer?)
- safety rules (read-only, row limits, timeouts)
- insight extraction (what does the result mean?)
- visualization rules (right chart for the data)

Great data analysis agents don’t pretend to answer everything — they answer a defined set of questions safely and clearly.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Customer Support Agent Designer