AI Automation / No-Code Automation

Design optimal Airtable base structure — tables, fields, linked records, rollups — for any workflow.
Difficulty: Intermediate
Model: GPT-4 / Claude / Gemini
Use Case: Database Design, Data Modeling
Updated: May 2026
Why This Prompt Exists
Bad Airtable schema is the #1 reason no-code projects fail. People build spreadsheets, not relational databases — then hit limits, duplication, and impossible reporting.

You get:

- duplicate data across tables (sync nightmares)
- reporting impossible because data isn’t linked
- row limit hits because you stored everything in one table
- automation failures because field types are wrong
- rebuilding from scratch after 3 months of wrong schema

But good schema follows patterns:

- entities: what are the distinct objects? (clients, projects, tasks)
- relationships: how do entities connect? (one-to-many, many-to-many)
- fields: what attributes does each entity have?
- rollups: what aggregated data needs to be visible?
- attachments: where do files live?

Without schema design, you build on quicksand.

This prompt designs optimal Airtable schemas for any workflow.

The Prompt

```
Assume the role of a database designer who specializes in Airtable.

Your task is to design an optimal schema for a given workflow.

Generate:

1. ENTITY IDENTIFICATION
   - List of entities (nouns in your workflow)
   - Example: "Client, Project, Task, Invoice"

2. RELATIONSHIP MAP

| Entity A | Relationship | Entity B | Cardinality |
|----------|--------------|----------|-------------|
| Client | has many | Projects | one-to-many |
| Project | has many | Tasks | one-to-many |

3. TABLE STRUCTURE (per entity)

**Table: [Entity Name]**
- Primary field: [Name/ID field]
- Fields:
  | Field Name | Type | Purpose | Required? |
  |------------|------|---------|-----------|
  | [name] | [Single line text / Number / Date / Link] | [description]
```

| Yes/No |

4. LINKED FIELDS
   - Which tables link to which? (Foreign keys in Airtable)

5. ROLLUP FIELDS
   - What aggregated data should be displayed? (e.g., "Total project tasks" on Project table)

6. LOOKUP FIELDS
   - What data should appear from linked tables?

7. VIEW RECOMMENDATIONS
   - Default view: [fields to show, sort order, filters]
   - Additional views: [by status, by owner, by date]

8. SCALING NOTES
   - Which table will hit row limits first? (and what to do)

INPUTS:

Workflow description:
[E.G., "Track client projects, tasks, deadlines, and invoices"]

Volume estimates:
[E.G., "500 clients, 10 projects per client, 20 tasks per project"]

Key reports needed:
[E.G., "Project status by client, late tasks, unpaid invoices"]

```
Existing data (if migrating from spreadsheets):
[PASTE SAMPLE ROWS]

RULES:
- One table per entity (avoid repeating data across tables)
- Use linked records for relationships (not duplicate text fields)
- Add rollup fields for reporting (avoid manual calculations)
- Set reasonable field limits (don't create 100 fields per table)
- Plan for 100K row limits (know which tables will grow fastest)
- Use formula fields instead of manual data entry when possible
```

How To Use It

- One table per distinct entity — if you're repeating data, you need a new table.
- Use linked records for relationships, not duplicate text fields (that's spreadsheet thinking).
- Add rollup fields for reporting — don't manually sum data across tables.
- Know the row limits: 100K for Pro plans, 250K for Enterprise.
- Plan for which table will hit limits first and have a migration path.

Example Input

Workflow description:
"Track course students, their progress through modules, assignment submissions, and grades."

Volume estimates:
"500 students, 20 modules, 5 assignments per module"

Key reports needed:
"Student completion rate, average grade by module, late submissions"

Existing data:
“Samples from Google Sheets with student names, module names, grades in separate columns”

Why It Works
Most Airtable users treat it like Google Sheets — one giant table with repeated data. That fails at scale.

This framework improves outcomes by forcing:

- entity identification (what are the distinct objects?)
- relationship mapping (how do entities connect?)
- field type selection (text vs. number vs. linked record)
- rollup and lookup planning (avoid manual aggregation)
- scaling awareness (which tables will hit limits)

Failure modes this prevents:

- 100K row limit hit because all data in one table
- Linked records set up backward (can’t roll up correctly)
- Formula errors from wrong field types
- Impossible reporting because no rollup fields

This improves on: Common internet advice to “use Airtable” without relational modeling. Most tutorials show simple lists, not multi-table schemas.

Related to: NCA-01 (Stack Selector) — this assumes Airtable is the right choice; NCA-03 (Make) for automations between these tables.

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 Make.com Scenario Architect