

Airtable Automations & Advanced Views — Workbook

This workbook turns the course into a build. Each section maps to one module and mixes hands-on exercises, fill-in worksheets, and checklists so you construct a real relational base, summarize it with rollups, automate the busywork, and ship an Interface as you go. Work in a fresh base built from a workflow you actually run, and by the end you will have a published, automated Airtable app instead of notes.

Relational Data With Linked Records

Convert a flat sheet into connected tables using linked records, a junction table, and a self-link.

Exercise: Split a flat sheet into linked tables

Take a real spreadsheet you maintain with repeated values (for example a project tracker where the client name is retyped on every row). Create two tables, add a Link to another record field connecting them, and move each repeated value into its own parent record so it lives in exactly one place.

- Which column in your old sheet was repeated, and which table does it now live in once?

-
- Name the two tables and the linked-record field that connects them.

-
- Did you limit the link to a single record or allow multiple? Justify the choice.

-
- Edit one parent value once. Confirm every linked child reflects it, and note what changed.

Worksheet: Table and relationship map

List every table your base will need and the links between them before you build. This is your schema on one page.

Table name

Primary field (what names each record)

Links to which other table(s)

Relationship type (one-to-many / many-to-many / self)

Single or multiple records per link

Key non-link fields this table owns

Exercise: Build a junction table for a many-to-many

Find a many-to-many in your data (students/classes, orders/products, staff/shifts). Create a junction table with a single-record link to each side, plus at least one detail field that describes the pairing (grade, quantity, status).

- What are the two sides, and what is the junction table named?

- Which detail field describes the pairing itself rather than either side alone?

- Create three pairing records. From one parent, can you see all its partners and their details?

Checklist: Clean data model check

- Every fact is editable in exactly one table (no retyped values)
- Parent links that must be unique are limited to a single record
- Each many-to-many uses a junction table, not two multi-link fields
- Any same-kind hierarchy uses a self-link with one parent per record
- Reverse link fields are renamed to read clearly (e.g. Subtasks)

Summarizing With Rollups and Lookups

Turn connected records into live answers with rollups, lookups, counts, and reactive formula fields.

Exercise: Roll up child values onto the parent

On a parent table, add a rollup that summarizes a child field across the link (for example SUM of project budgets on each client). Then add a conditional rollup that only includes children meeting a condition (for example only Active projects).

- Write the rollup's linked field, child field, and aggregation function (SUM / AVERAGE / COUNTA / MAX).

- What condition did you add to the conditional rollup, and how did the number change?

- Add a new child record. Confirm the parent total updates with no manual action.

- If a rollup showed blank, what was the cause and how did you fix it?

Worksheet: Summary-field planner

For each answer you want on a parent record, choose the right field type before building. Match the question to lookup, rollup, or count.

Question to answer on the parent (e.g. total budget, item count, client email)

Field type (Lookup / Rollup / Count)

Linked field used

Child field summarized (if any)

Aggregation function (rollup only)

Filter condition on included records (if any)

Exercise: Add a self-grading health formula

Using a rollup of incomplete children and a date field, write a formula field that returns At Risk, On Track, or Complete. Use IF/AND with DATETIME_DIFF and TODAY so the label updates automatically.

- Write the full formula you used for the Health field.

• Which functions did you use (IF, AND, DATETIME_DIFF, SWITCH)?

• Finish a child task so the rollup hits zero. Did the label flip to Complete on its own?

• How did you make an empty rollup behave (blank vs zero) so the formula stayed correct?

Checklist: Live-summary readiness

Each summary field uses the right type: lookup for a value, rollup for math, count for how-many

Currency and number rollups are formatted with correct decimals

Conditional rollups include only the records that should count

Formula fields turn rollups into status labels that update automatically

Blank rollups are handled so formulas do not misread empty as an error

Automations That Run the Work

Make the base act on its own with trigger-action automations, record updates, and a scripting step where needed.

Exercise: Build a condition-based alert

Create an automation using the When a record matches conditions trigger (for example Priority is High). Add a Slack or email action with dynamic tokens for the record name and link. Test it on a real sample record, then turn it on.

- What trigger and exact condition did you use?

• Which dynamic tokens did you insert so the message is specific, not generic?

• What did the test message look like for the recipient?

• Why is When a record matches conditions usually safer than a raw creation trigger here?

Worksheet: Automation design sheet

Spec each automation before building so triggers do not collide or loop. Fill one block per automation.

Automation name

Trigger type and condition

Action 1 (notify / update record / create record)

Action 2 and any conditional branches

Records or links it writes to

Loop risk: does any field it sets trigger another automation?

Exercise: Auto-update and auto-create across tables

Build an automation that, when a status changes (for example Stage becomes Won), updates the source record (stamp a date, change stage) and creates a linked record in another table (a kickoff task) using the trigger record's ID for the link.

- Write the trigger condition and the fields your Update record action sets.

• What record does the Create record action add, and in which table?

• How did you link the new record back to its source (which token did you use)?

• Confirm the new record appears under the parent's rollup. Did the link hold?

Checklist: Automation safety and limits

- Every automation was tested on a real sample before being turned on
- Triggers are specific enough that the automation only fires when it should
- No two automations set fields that re-trigger each other (no loops)
- Created records are linked back to their source to preserve relationships
- Total runs are within the plan's monthly automation-run limit

Advanced Views and Interface Designer

Make the base usable for everyone with grouped/filtered/colored views, visual layouts, and a published Interface.

Exercise: Build three audience-specific views

On one table, create three saved views for three audiences: a personal My Open Items (filtered to current user and not done), a manager view grouped by Stage and sorted by Due Date, and an Overdue view with conditional record coloring turning late, unfinished items red.

- Write the filter conditions for each of the three views.

• Which field did you group the manager view by, and what do the group summaries show?

• Write the conditional coloring rule that turns overdue, unfinished records red.

• Confirm none of the view changes altered the underlying data.

Worksheet: View catalog

Document every view so the sidebar reads like a menu of reports for your team.

View name (audience + question)

View type (Grid / Kanban / Calendar / Gallery)

Filter conditions

Group by / Sort by

Color rule

Locked? (shared views)

Exercise: Ship a dashboard interface

In Interface Designer, build one page with at least two number cards bound to rollups, one chart (records grouped by status), a filtered list, and a record-detail page that exposes only chosen editable fields. Share it with one person who does not have full base access.

- Which rollups did your number cards display, and what does the chart group by?

- On the record-detail page, which fields are visible and which are editable?

- Who did you share it with, and what can they NOT see or edit?

- Make an edit inside the interface. Confirm it updated the table, view, and rollups.

Checklist: Ship-the-app checklist

- Chose one real workflow and modeled its base with proper links
- Added rollups, lookups, and a health formula so records summarize themselves
- Built at least one alert and one update/create automation, both tested
- Created audience-specific views including a visual Kanban or Calendar
- Published an Interface and shared it with a non-admin user who only sees what they need

Your Action Plan

1. Pick one real workflow (intake, projects, CRM, inventory) and sketch its tables and links on the relationship map.
2. Build the tables, add linked-record fields, and split every repeated value into its own parent record.
3. Add a junction table for any many-to-many and a self-link for any same-kind hierarchy.
4. Add rollups, lookups, and count fields so each parent record summarizes its children live.
5. Add a formula field that grades each record's health from its rollups and dates.
6. Build one condition-based alert automation with dynamic tokens, test it, and turn it on.
7. Build one update-and-create automation that advances work across tables, linking new records back.
8. Create three audience-specific views plus one Kanban or Calendar for visual work.
9. Assemble a dashboard and record-detail page in Interface Designer and publish it.
10. Share the interface with a non-admin teammate, then review automation runs against your plan limits monthly.

