STRETCHLEARN · WORKBOOK

Airtable for Everyday Organization — Workbook

This workbook turns the course into the real modeling decisions and building work that make Airtable run your projects, inventory, contacts, and business data from one connected base. Each section maps to a course module: thinking in tables and planning your data model, mastering field types, linking tables and building daily views, and automating and sharing so the base maintains itself. Do the work in Airtable as you go, because a database is only learned by building one. The included spreadsheet and document templates are ready to import into Airtable or keep as a planning reference, so you never have to rebuild your structure from memory.

Airtable Foundations and Thinking in Tables, Not Tabs

Plan your data model on paper, then build your first base and tables so the relational structure is right before any data goes in.

Exercise: Model Your Data on Paper First

Before opening Airtable, write down the real-world things you manage and how they relate, using the planning sentence each [A] has many [B]. This ten-minute step is what prevents a base you have to rebuild.

• List every repeating noun in the area you want to manage (e.g. clients, products, projects, suppliers, orders, members).

• For each noun, write down the details it owns that no other noun owns (e.g. a client owns email, phone, rate, status).

• Write each relationship as a sentence in the form each X has many Y (e.g. each supplier has many products).

• Convert your sentences into a table list: each noun becomes a table, and note which tables will link to which.

Worksheet: Base and Table Plan

Capture the decisions that define your base before you build it. Fill every field; the gaps you cannot fill are exactly the decisions to make now rather than mid-build.

Name of this base and the single area of life or business it covers

The tables this base will contain (list 2-5 tables)

Primary field for each table (the human-readable name shown when linked)

Relationships between tables, written as each X has many Y

The one question I want this base to answer at a glance (e.g. what orders ship today)

Airtable plan I am using (Free, Team) and the record cap I must stay under

Exercise: Build the Skeleton in Airtable

Open Airtable, create the base from scratch (not a template), and build the empty structure from your plan. Add real records but no fancy fields yet; the goal is a clean skeleton you will flesh out in later sections.

• Create a base named from your plan and rename Table 1 to your first real table.

• Set each table's primary field to the human-readable name you chose (a name, not an ID).

• Add every table from your plan and enter three or four genuine records in each.

• Open one record using the expand arrow to view it as a full card, where links will later appear.

Checklist: Foundations Ready Checklist

[ ] I wrote my data model on paper before building, using each X has many Y sentences.
[ ] My base is named for one connected area, with related tables inside it (not split across bases).
[ ] Every table's primary field is a clear human-readable name, not an ID or code.
[ ] Each table has a few real records so I have genuine data to shape.
[ ] I know my plan's record cap and roughly how close my base is to it.

Mastering Field Types, the Real Power of Airtable

Assign the right field type to every column, add computed fields, and audit your tables so the data stays clean as it grows.

Worksheet: Field Type Plan for Each Table

For your most important table, decide the type of every field before changing it in Airtable. Picking the type deliberately is what keeps filters, sorts, and totals working later.

Table I am planning fields for

Single select fields and their fixed option lists (e.g. Status: Pending, Paid, Packed, Shipped)

Number or Currency fields (e.g. Unit Price, Quantity, Order Total)

Date fields and whether they need a time (e.g. Due Date, Order Date)

Email, Phone, URL, and Attachment fields

Any field currently stuffing two facts into one cell that I must split

Exercise: Add Computed Fields That Update Themselves

Replace at least one manually maintained number with a Formula field so it recalculates on its own. If your tables are linked already, add a rollup or lookup too; if not, note it to add after Section 3.

• Add a Formula field for a line total using {Quantity} * {Unit Price}, or a similar calculation in your base.

• Add a Formula field that produces text or a flag, such as IF on days remaining to label work Overdue or On track.

• Plan one lookup field that will pull a value across a link (e.g. show a client's rate inside Projects).

• Plan one rollup field that will aggregate linked records (e.g. sum a client's total project value).

Worksheet: Single Select Option Design

Design the fixed option lists that replace free text. Consistent options are the single biggest fix for messy data, so choose them carefully and give each a color.

Status options in order and their colors (the pipeline stages records move through)

Priority options (e.g. High, Medium, Low) and their colors

Category or Type options for grouping records

Multiple-select tag options for cross-cutting labels (e.g. Bestseller, Seasonal)

My rule for adding a new option versus reusing an existing one

Checklist: Clean Fields Checklist

[ ] Every column has the correct field type, not defaulted to text.
[ ] Status, priority, and category use single select with fixed colored options, not free text.
[ ] Numbers and prices use Number or Currency so they sort and sum.
[ ] No field crams two facts into one cell; each fact has its own typed field.
[ ] At least one manually updated number is now a self-calculating Formula field.
[ ] Rarely used fields are hidden in the view, not deleted, and key fields have descriptions.

Linking Tables and Building Views You Check Daily

Connect your tables with linked records, then build the Grid, Kanban, Calendar, and Gallery views and dashboards you will actually open every day.

Exercise: Wire Your Tables Together With Linked Records

Turn your separate tables into a system by adding linked-record fields from your Section 1 plan. Confirm the reverse links appear and that changing data in one place flows everywhere.

• Add a Link to another record field connecting two tables (e.g. Projects linked to Clients).

• Open a record and confirm the automatic reverse-link field appears in the other table.

• For one-to-one relationships, turn off Allow linking to multiple records so each links to exactly one.

• Edit a linked record's detail (e.g. a client's email) and confirm it updates everywhere it is referenced.

Worksheet: View Plan: One Table, Many Lenses

For one important table, plan the views you need so each job gets a screen built for it. Naming each view for its purpose now keeps your sidebar usable later.

Table I am building views for

Grid view name and which fields it should show for editing

Kanban view name and the single select field it stacks by (e.g. Status)

Calendar view name and the Date field it plots (e.g. Due Date or Publish Date)

Gallery view name and the attachment field used as the cover image

The one filter each view needs to show only its relevant records

Exercise: Build a Working Dashboard View

Combine filters, sorts, and grouping in a single view to answer a question you ask often. The result should be a screen you can open daily and instantly trust.

• Create a view and add a filter for its job (e.g. Status is not Shipped, or Due Date within 7 days).

• Add a sort, such as Priority descending then Due Date ascending.

• Group the view by a single select (e.g. Status or Category) so records cluster under headings.

• Turn on a group summary to show a count per group or a Currency sum in each group header.

Checklist: Connected and Visible Checklist

[ ] My tables are linked with linked-record fields, and reverse links appear automatically.
[ ] Changing a fact in one record updates it everywhere it is linked (no copied data left).
[ ] Each important table has multiple views matched to their jobs (Grid, Kanban, Calendar, or Gallery).
[ ] At least one view uses a filter so it shows only the records relevant to its purpose.
[ ] I have one dashboard view that combines filter, sort, and grouping to answer a daily question.
[ ] Where tables are linked, I added the lookup and rollup fields I planned in Section 2.

Automations, Interfaces, and a Base That Runs Itself

Automate the repetitive parts, build a clean Interface to share instead of the raw grid, and set up the import, export, and maintenance habits that keep the base trustworthy.

Exercise: Build Your First Automation

Create one automation that removes a recurring chore. Use the Test step to run it against a real record and confirm the output before turning it on, so you never unleash a broken rule on the whole base.

• Pick the chore to automate and write it as when this happens, do that (e.g. when Status is Shipped, email the customer).

• Build it with a trigger of When a record matches conditions and at least one action (update record, send email, or create record).

• Map fields from the triggering record into the action (e.g. the linked customer's email and the order total).

• Run the Test step with a real record, confirm the preview, then turn the automation on.

Worksheet: Sharing and Interface Plan

Decide exactly who sees what before sharing anything. Matching access to need is how you put a base in front of clients or teammates without exposing your workshop.

Each audience for this base (e.g. me, a teammate, a client) and what they actually need to see and do

Permission level for each collaborator (Creator, Editor, Commenter, Read-only)

Which audience needs an Interface instead of the raw base, and what elements it should contain (list, detail panel, charts)

Fields that must stay hidden from shared views (formulas, rates, supplier data)

Which views, if any, I will share by link and at what access level

Worksheet: Import, Export, and Maintenance Plan

Plan how data comes in, goes out, and stays clean over time. A base you tend monthly is a base you still trust a year from now.

Existing data I will import and the field types I must fix after import (text to single select, text to Currency, text to linked record)

Which views I will export to CSV and why (sharing, reporting, feeding another tool)

How and how often I will back up the base (e.g. duplicate the whole base monthly)

My monthly maintenance steps (archive finished records, fix field-type drift, tidy select options)

How I will watch the record count against my plan's cap so I upgrade before hitting it

Checklist: Self-Running Base Checklist

[ ] At least one automation removes a recurring chore and was tested before going live.
[ ] I am mindful of my plan's monthly automation-run limit and reserve runs for useful triggers.
[ ] Outside viewers see a clean Interface, not the raw base with its formulas and reference tables.
[ ] Collaborators are invited at the lowest permission level that lets them do their job.
[ ] Imported data had every field type verified, and name columns were converted to linked records.
[ ] I have a backup habit and a monthly maintenance loop, and I watch the record count against my cap.

Your Action Plan

1. Write your data model on paper using each X has many Y, then create a base from scratch with one table per noun and a human-readable primary field for each.
2. Enter real records, then set the correct field type for every column: single select for status and category, Currency for prices, Date for deadlines.
3. Replace at least one manually updated number with a Formula field so it recalculates itself.
4. Add linked-record fields to connect your tables, and confirm the automatic reverse links and that edits flow everywhere.
5. Add lookup fields to pull values across links and rollup fields to aggregate linked records (e.g. total revenue per client).
6. Build the views each job needs from one table: a Grid for editing, a Kanban stacked by Status, a Calendar on your Date field, a Gallery with cover images.
7. Create one dashboard view that combines a filter, a sort, and grouping with a group summary to answer a question you ask daily.
8. Build and test one automation (e.g. when Status changes to Shipped, email the customer), then turn it on.
9. Build an Interface for anyone outside your team so they see a clean page, and invite

collaborators at the lowest permission that fits.
10. Set up a backup habit, verify field types on any imported data, and schedule a monthly maintenance loop while watching your record count against the plan cap.