



The Jewish Nonprofit's Guide to Notion

The whole product, from the atom up — blocks, pages, databases, formulas, AI, automations, permissions, cost, stewardship, and a rollout plan

Free to read, keep, and forward

Shalom Desk · a Notion workspace for Jewish community organizations
Jacob Sager · Brand New Colors · jacobsager.com

Everything in Notion is a block.

Notion is a workspace app used by more than a hundred million people, and its entire design rests on one small idea: every piece of content is a block. A paragraph is a block. A heading, a checklist item, an image, a table, an embedded map, a quote — each is a block, and blocks are interchangeable, movable, and transformable. Hover beside any block and a  handle appears: drag it anywhere on the page, drop it beside another block and the page splits into columns, click it for the menu that turns one block type into another.

Type a forward slash anywhere and the full block menu appears at your cursor — `/heading`, `/todo`, `/callout`, `/table` — which means building a page never requires leaving the keyboard. Markdown shortcuts work as you type: a dash and a space makes a bullet, `##` makes a heading, `[]` makes a to-do, `>` makes a quote. The consequence takes a week to sink in: a Notion page has no fixed shape. The meeting-notes page that started as a plain document grows a checklist, then a decisions table, then a linked calendar — without ever opening another app. Documents and structure stop being separate categories.

The block types a congregation actually uses

BLOCK	WHAT IT'S FOR IN YOUR WEEK
Text · Headings 1–3	The writing itself, and the structure that makes long pages scannable
To-do list	Checklists that hold state — the event setup list that remembers what's done
Toggle · Toggle heading	Collapsible sections; the detail hides until someone needs it
Callout	The colored attention box — warnings, welcomes, the note everyone must read
Bulleted / numbered lists	Everything meeting minutes are made of
Table (simple)	Quick grids without database machinery — a phone tree, a service schedule
Table of contents	Auto-built navigation for any page with headings — handbooks, policies
Quote · Divider	Structure and breathing room for longer pages
Image · File · PDF · Video	The flyer, the signed contract, the building map — attached where they belong
Embed	Live windows on outside things: a Google Map to the cemetery, a Drive doc, a form
Columns	Side-by-side layout, made by dragging one block next to another
Synced block	One block appearing in several places, editing everywhere at once — office hours listed on three pages, corrected in one
Linked view of a database	A window into a database placed on any page — the mechanism dashboards are made of

Transformation is the quiet superpower: any block becomes any compatible block from its handle menu. The brainstorm list becomes a to-do list becomes the headings of the actual plan, with nothing retyped. Blocks are also the unit of selection — grab three, drag them into a toggle, and the page tidies itself.

The shortcuts worth memorizing in week one

Nearly everything above can be typed rather than hunted for. These are the ones that pay for themselves the first afternoon:

TYPE THIS	AND YOU GET
/	The block menu, right at your cursor — /table, /callout, /toggle, /date
# · ## · ### + space	Heading 1, 2, or 3, formed as you type
- + space	A bulleted list item
1. + space	A numbered list that keeps counting
[]	A to-do checkbox
> + space	A toggle that collapses everything nested inside it
@	A person (notifies them), a page (links it), or a date (reminds you)
bold · <i>*italic*</i>	Formatting without lifting your hands — or Cmd/Ctrl + B and I
Tab · Shift+Tab · Esc	Indent, outdent, or select the whole block to drag it

Try it now: build a real page in four moves

Reading about blocks teaches less than five minutes of using them. Make something the office will actually keep:

- 1 Create the page**
In the sidebar, hover a section and click +. Name it Board Meeting — March, then pick an icon above the title.
- 2 Type the structure**
Type ## Agenda and press enter — the heading forms itself. Underneath, type a dash and a space, then list the three things the meeting must cover, one per line.
- 3 Make it hold decisions**
Below the agenda, type ## Decisions, then [] to open a checkbox. Write one decision and @-mention the person who owns it; they will be notified and it will appear in their Home.
- 4 Rearrange without retyping**
Drag any block by its ■■ handle, then use that same menu to turn an agenda bullet into a to-do — the list you wrote becomes the list you work from.

Why this exercise and not a features tour

That little page is the pattern behind everything later in this book: blocks typed as you go, ownership assigned by mention, nothing retyped when the shape changes.

Pages nest, and the sidebar is the building.

A page is a block that holds other blocks — including other pages. Nesting is how a workspace takes shape: a Board page holds a page per meeting; a High Holidays page holds the service plans, the volunteer roster, and last year's debrief. There's no separate folder system to fight, because pages inside pages are the folders — and any page can carry an icon and a cover image, so the important ones look like themselves. Breadcrumbs at the top of every page show where you are; Favorites pin the pages you live in to the top of the sidebar.

Reading the sidebar

- **on team plans, top-level sections with their own membership — one per functional area maps cleanly onto a congregation: Membership, Finance, Events, Board. A person sees the teamspaces they belong to and nothing else.** — Teamspaces
- **pages shared directly with you outside a teamspace — the consultant's project page, the committee doc.** — Shared
- **a genuinely personal section no one else can see. Drafts belong here until they're ready; moving a page from Private to a teamspace is publication.** — Private
- **global search reaches everything you can access, including inside databases. Home surfaces recent pages and your assigned tasks, and becomes the front door most staff use daily.** — Search and Home

Links, mentions, and backlinks

Type @ to mention a page, a person, or a date anywhere in text. Mentioning a person notifies them; mentioning a date can carry a reminder (@tomorrow 9am pings you then); mentioning a page creates a live link that survives renames. Every page quietly lists its backlinks — the other pages pointing at it — so the knowledge web assembles itself out of ordinary writing. The how-we-do-things page ends up findable from every place it matters, which is the whole point of writing it down.

A habit worth forming in week one

When information arrives — a vendor quote, a decision, a phone number — give it a page or add it to one, and link it from where people will look. The office that writes into the workspace stops asking whether anyone saw the email.

Comments, tasks, and the end of reply-all.

- highlight any text and comment on it; comment at the top of a page for the general thread. Discussions resolve when they're done, leaving the page clean and the history kept. The draft bulletin stops circulating as attachments — it lives at one URL, and the edits argue politely in the margin. — Comments and discussions
- an @-mention inside a to-do effectively hands it to someone, and person properties on database records make ownership explicit. Assigned items surface in each person's Home, so Sunday's volunteer sees Sunday's list without anyone forwarding it. — Mentions that assign
- Notion collects your mentions, comments, and page invitations in one inbox. Checked each morning, it replaces a surprising share of internal email; ignored, it's ignorable — the pages themselves stay authoritative. — The notification inbox
- @-date mentions with times ping the right person at the right moment — @March 1 on the line about the oven inspection is the cheapest tickler system ever built. — Reminders

A worked scenario: this month's bulletin

- 1 One page, not eleven attachments**
The volunteer creates Bulletin — March and writes the draft. That page's URL is the bulletin now; it never becomes v3_final_FINAL.
- 2 Everyone reacts in the margin**
The rabbi highlights a paragraph and comments; the office manager comments on the dates. Each thread sits beside the text it concerns.
- 3 Fixes land in place**
Whoever is right edits the line and resolves the thread, with the history kept. Nobody merges anything by hand.
- 4 The decision is captured**
A to-do at the bottom, @-mentioned to whoever sends it, gives the last step an owner instead of an assumption.

Where notifications come from

YOU GET PINGED WHEN	BECAUSE
Someone @-mentions you	They wanted your eyes on this line specifically
Someone comments on a page you follow	You edited or subscribed to it
A reminder date arrives	Someone typed @March 1 with a time

The habit that replaces reply-all

When a discussion needs more than two exchanges, move it onto the page it concerns. Email keeps the conversation; the page keeps the conclusion — and only one stays findable in a year.

Records are pages with fields attached.

A database is Notion's structured heart, and the concept that separates it from every notes app: each row is a full page — open it and there's room for notes, checklists, and history — while its properties, the fields attached to it, are what views can sort, filter, group, and count. A membership directory, a donations log, and an event calendar are one idea wearing three outfits: databases whose records carry different properties.

The property types, and when each earns its place

PROPERTY	HOLDS	CONGREGATION EXAMPLE
Text · Number	Free words · amounts and counts	Notes; Annual Dues formatted as dollars
Select · Multi-select	One choice · several, from a defined list	Dues level: Full / Sustaining / Reduced; dietary needs
Status	A staged workflow with To-do / In-progress / Done groups	Invoice: Draft → Sent → Paid
Date	A date or range, optionally with time and reminder	Event date; membership term
Person	A workspace member	Staff owner of a program
Checkbox	Yes or no	Acknowledgment sent; attendance present
Email · Phone · URL	Contact fields that behave like contact fields	The directory's working columns
Files & media	Attachments living on the record	The signed rental contract on the booking
Relation	A live two-way link to records in another database	This donation ↔ that family
Rollup	Math across a relation	A family's total giving, computed live
Formula	Logic across this record's own properties	Attendance % from sessions present over recorded
Created / edited time & by	Automatic housekeeping	Who touched the record, and when
ID	An auto-numbered identifier	Invoice numbers that assign themselves

Two structural extras matter as databases mature: sub-items let records contain child records (a project with its tasks), and dependencies connect records in sequence for timeline planning — useful the year the renovation project needs a real schedule.

Views: one truth, many faces.

The same records can be shown as a table for auditing, a board grouped by status for triage, a calendar or timeline for scheduling, a gallery of cards for browsing, a list for reading, and a chart for the trend the board asks about. Views carry their own filters, sorts, and groupings — the treasurer's view of donations and the rabbi's view of families are different windows onto identical records, and none of them are copies. Copies are how truth splits; views are how it multiplies safely.

- **conditions combine with and/or into groups: Status is Member AND (Last Touchpoint before 60 days ago OR no touchpoint at all). Build the question once and the view keeps answering it forever.** — Filters, compound
- **group a table by any property to see structure — families by status, gifts by fund — and the calculation row at the bottom of any column sums, averages, counts, or percents on demand. The month's giving total is a click, not a formula.** — Grouping and calculations
- **when a dashboard view is right, lock it (■ → Lock views) so a stray drag can't rearrange what the whole office relies on. Databases themselves can lock their schemas, which keeps helpful volunteers from inventing properties.** — Locked views

Database templates and forms

A database template is a pre-built page for new records — the new-family intake that arrives with its checklist written, the case note that opens with its confidentiality banner in place — and templates can repeat on a schedule, creating each week's service record before anyone asks. Notion forms go one step outward: publish a form whose submissions become records directly, and the event sign-up sheet finally is the registration database instead of something a volunteer retypes into it on Monday.

Which view for which job

VIEW	SHOWS RECORDS AS	REACH FOR IT WHEN
Table	Rows and columns	Auditing, bulk edits, seeing many properties at once
Board	Cards in columns by a property	Triage — invoices by status, cases by stage
Calendar	Cards on dates	Programs, services, anything the staff schedules around
Timeline	Bars across a date range	Multi-week projects: the renovation, the capital campaign
Gallery	Cards with a summary or image	Browsing people and programs; the friendliest for volunteers
List	A quiet single-column stack	Reading — policies, meeting notes, documents
Chart	Bars, lines, or donuts	The giving trend the board asks about every month



One database usually carries three or four views, each built for a different person's job. Adding a view costs nothing and changes nothing underneath — which is why the answer to "can I see this differently?" should always be yes.

Relations and rollups — where lists become a system.

A relation links records across databases, and the link runs both ways: connect a donation to the Goldberg family and the family's record lists the gift while the gift shows its donor. Enter the fact once and every surface that cares about it stays current, retiring the entire category of work called keeping the lists in sync. A rollup then does math across the relation — sum the linked donations and the family's lifetime giving computes itself; take the latest linked touchpoint date and the quiet households become visible.

Worked example one: giving

- 1 Two databases**
Families and Donations, each holding its own kind of record — one per household, one per gift.
- 2 One relation**
A Donor property on Donations pointing at Families; Notion creates the reverse automatically, so families list their gifts.
- 3 One rollup**
Total Giving on Families: read the Donor relation, take each gift's Amount, sum. It updates the moment any gift is added or edited.
- 4 The payoff**
The board report's biggest number now exists before the meeting is scheduled — and it's right, because it was never a copy of anything.

Worked example two: the event that counts itself

- 1 Two databases**
Events and Registrations. Each event carries a Capacity number; each registration carries a Party Size.
- 2 One relation**
Registrations point at their Event.
- 3 Two computed properties**
A rollup on Events sums party sizes into a headcount; a formula subtracts it from Capacity into Seats Remaining.
- 4 The payoff**
The question is there room becomes a number on the card, counting itself down as households sign up — and the waitlist conversation happens before the room is oversold, never after.

The pattern to remember

One kind of thing per database, relations for every real-world connection, rollups where a number should compute instead of being typed. Systems built this way get smarter with every record entered; systems built as one giant table get heavier.

Five formulas that earn their keep.

Formulas compute within a record, using its own properties. The language reads like a spreadsheet's, with functions for dates, text, and logic. Five patterns cover most of what a community office wants:

THE WANT	THE FORMULA	READS AS
Age from a birthdate	<code>dateBetween(now(), Birthdate, "years")</code>	Years between now and the birthdate
Days until an event	<code>dateBetween(EventDate, now(), "days")</code>	Countdown that updates daily
A balance label	<code>if(Balance > 0, "Outstanding", "Clear")</code>	Words instead of a number to squint at
Attendance percent	<code>round(Present / Recorded * 100)</code>	Sessions attended over sessions held
A card summary	<code>Status + " · \$" + format(Dues)</code>	One readable line for gallery cards

The last pattern deserves adoption everywhere: a Summary formula composing a record's key facts into one line turns every gallery into cards that read like sentences — and it maintains itself, because changing the dues corrects the card.

Four more that solve real office problems

THE WANT	THE FORMULA	READS AS
Next renewal date	<code>dateAdd(Start, 1, "years")</code>	One year on from the term's start
Flag missing contact info	<code>if(empty(Email), "Missing", "OK")</code>	A hygiene column for directory cleanup
Days overdue	<code>dateBetween(now(), Due, "days")</code>	Positive numbers are the follow-up list
Days until an event	<code>dateBetween(EventDate, now(), "days")</code>	A countdown that refreshes itself daily

Working with the formula editor

- **start simple, confirm on one record, then add the logic you want.** — Build it in pieces
- **describing the result in plain language beats hunting through function documentation.** — Let AI write the first draft
- **Renewal Due helps the office; dateAdd_start_1y helps nobody.** — Name the output, not the mechanism

When a formula is the wrong tool

Formulas compute within one record. The moment a number must reach across records — a family's total giving, an event's headcount — the answer is a rollup instead.

An assistant that already knows your filing system.

Notion AI works inside the workspace, reading and writing the same pages and databases you do — the property that makes it different from a chatbot in another tab. Its abilities cluster into four kinds of help, each mapping directly onto a community office's week:

- **draft, rewrite, summarize, translate, adjust tone — on any page, from the bulletin blurb to the condolence note that deserves a careful second pass. Highlight text and ask; the suggestion arrives in place.** — Writing and editing
- **question the workspace in plain language — which families gave to the Building fund this quarter, what did we decide about the kitchen renovation — and get answers drawn from your own pages and databases, with sources shown. On business tiers the search extends across connected tools, so the answer that lives in a Slack thread surfaces too.** — Ask, and search that answers
- **AI-filled properties summarize each record's page into one line, categorize entries, extract details, or tidy a messy import — the migration cleanup that used to eat a volunteer's weekend runs as a column.** — Database autofill
- **hand it a task that spans pages and databases — build a view of lapsed memberships and draft the renewal letter, clear the sample data from every database — and it executes the sequence, reporting as it goes. AI meeting notes transcribe and summarize the meetings themselves.** — The Agent, for multi-step work

What the AI costs, honestly

As of mid-2026: full Notion AI — the Agent, AI search, autofill, and meeting notes — ships inside the Business plan at \$20 per member per month billed annually (\$24 monthly). Free and Plus carry a limited trial rather than the full toolset; the old standalone AI add-on was retired in 2025. Separately, custom agents meter through credits at \$10 per thousand. The practical read for a small office: run the workspace on Plus, and buy Business seats only for the one or two people who'll genuinely work with the AI daily.

Prompts that earn their keep in a shul office

Summarize this meeting page into decisions and owners · Draft a thank-you letter from this donation record · List member families with no contact in 90 days · Turn last year's Purim debrief into this year's checklist · Categorize these imported gifts by fund · Clear the sample data from the Events database only.

A word on judgment

The AI drafts; a person sends. Condolences, sensitive care notes, and anything touching a family's dignity deserve human final passes as policy, written down where the whole staff can see it. The tool is excellent company and a poor final authority — which is true of every tool a sacred community uses.

The workspace that does its own housekeeping.

Database automations watch for a trigger and perform actions, no code involved. Triggers include a page being added to the database or a property changing — any status flip, any checkbox, any date arriving. Actions include editing properties, adding pages (in this database or another), notifying a person, posting to Slack, and firing a webhook toward outside tools. Buttons compress a routine into one click, on a page or as a property on every record — inserting blocks, creating records, editing pages, opening a confirmation first when the action deserves a pause. Recurring database templates create records on a schedule.

RECIPE FOR A CONGREGATION	TRIGGER → ACTION
Confirmations that confirm themselves	Registration's Paid checked → status becomes Confirmed, date stamped
Care hygiene	Pastoral case moved to Closed → the closed date fills itself
Money follow-through	Invoice status becomes Sent → treasurer notified; becomes Paid → date stamped
New-member welcome	Page added to Families → a welcome checklist page generates and assigns itself
One-click logging	A Log Touchpoint button creates today's contact record, cursor waiting in Notes
The week that builds itself	A recurring template creates each Shabbat's service record on schedule

Building your first one

- 1 Open the target database**
 Click the  icon (or  → Automations) and choose New automation.
- 2 Pick the trigger**
 The database's own properties appear as conditions — choose the checkbox, the status value, or page-added.
- 3 Stack the actions**
 Edit a property, stamp a date, notify a person. Actions run in order; two or three cover most office routines.
- 4 Name it, test it, trust it**
 Flip one record and watch it work. Automations live on paid plans — and the honest sequencing is to automate a habit the office already keeps manually, so the machinery serves rather than confuses.

One workspace, doors where they belong.

Access in Notion is granted page by page, and everything nested inherits unless overridden — exactly the shape a congregation needs. Members belong to the workspace; guests receive specific pages without joining at all (the auditor, the seasonal volunteer, the visiting consultant), and paid plans allow unlimited guests. Every grant carries a level — full access, can edit, can comment, can view — and the levels are the vocabulary of trust: the board can comment on the budget without being able to change it.

Teamspace group areas with their own membership; business tiers add private teamspace whose existence is invisible to non-members — the natural home for pastoral care and personnel files. A starting access map that fits most congregations:

AREA	FULL ACCESS	EVERYONE ELSE
Pastoral care	Clergy + one designated staff member	An open-case count at most — never contents
Giving & billing	Treasurer, bookkeeper, executive director	Totals on reports, not household histories
Member directory	Office staff and clergy	Board sees counts; volunteers see event lists
Board materials	Board and senior staff	Published minutes only, where you choose
Events & programs	Program staff + key volunteers	The public calendar, marked public on purpose
Institutional memory	All staff	Whatever each page's sharing says

- **any page can become a public webpage — the community calendar, the class schedule — while everything around it stays private. Publishing carries its own toggles: whether search engines may index it, whether visitors may duplicate it as a template. Publishing is a per-page decision, never a default.** — Publish to web, deliberately
- **the strongest pattern adds a confidentiality level to sensitive records themselves — High, Standard, Low — so authorized eyes also know how to treat what they read.** — Grade sensitivity on the record
- **sensitive areas get their permissions before their first real record. The pastoral note entered before the lock was set is a breach that can't be un-rung.** — Doors before furniture

Four plans, one discount, three worked examples.

PLAN	PRICE (ANNUAL BILLING)	WHAT IT ADDS
Free	\$0	Genuinely usable solo — unlimited pages for one person; capped collaborative blocks with a team
Plus	\$10 / member / month (\$12 monthly)	The small-team tier: unlimited collaboration, unlimited file uploads, 30-day page history, unlimited guests
Business	\$20 / member / month (\$24 monthly)	Full Notion AI (Agent, AI search, meeting notes), private teamspaces, SAML SSO, 90-day history, advanced permissions
Enterprise	Custom	Large-org controls: audit logs, provisioning, data residency, dedicated support

The number that matters most for this audience: registered 501(c)(3) organizations can apply for 50% off through Notion’s nonprofit program, verified through TechSoup — and most synagogues, JCCs, and Jewish day schools qualify. The process is a TechSoup validation of your organization, then the discount applied to your plan; apply before inviting the team and every seat lands discounted from day one.

The math for three real offices

- **three seats on discounted Plus at \$5 each: \$15 a month, \$180 a year — for the platform running membership, giving, events, and the institutional memory.** — The three-person shul office
- **six discounted Plus seats (\$30) and two discounted Business seats for the office manager and ED who'll use the AI daily (\$20): \$50 a month, \$600 a year.** — The eight-person staff with AI users
- **twenty-five discounted Plus seats: \$125 a month, \$1,500 a year — landing under a single year of typical dedicated-platform pricing, which runs \$1,600–\$2,400 annually before per-module fees.** — The 25-person day school

Getting it approved

- **a board that hears \$180 a year beside \$2,000 a year for comparable capability approves quickly. The same board asked to evaluate a workspace app deliberates for months.** — Lead with the comparison, not the feature list
- **if the workspace retires a dedicated subscription, a form service, or forty hours of annual reconciliation, say so in the same sentence as the price.** — Name what it replaces
- **annual billing with a stated review date reads as prudent rather than permanent, and the review almost never ends in reversal once the directory is live.** — Ask for one year, not forever
- **a two-minute walk through a family record — giving, dues, enrollment, and last contact on one screen — persuades more than any spreadsheet of features.** — Show them the thing



The number worth writing on the memo

Nonprofit-discounted Plus runs about \$60 per person per year. Frame the request in those terms and it lands beside a modest supply line rather than a software procurement.

Getting data in, keeping it safe, getting it out.

- Notion ingests CSVs (straight into databases, columns mapping to properties), Word documents, Evernote, and more. The membership spreadsheet becomes a database in one merge — and the cleanup that follows is exactly what AI autofill is for. — [Import](#)
- everything exports — CSV per database, or the entire workspace at once from settings. A monthly export stored with your other backups is cheap insurance, and the right answer when the board asks who holds our data: we do, in two places. — [Export, as a habit](#)
- page history lets you rewind edits (30 days back on Plus, 90 on Business), and deleted pages sit in Trash awaiting recovery. The honest fear — what if someone breaks it — has a short answer: restore it. — [History and the trash](#)
- when a spreadsheet retires, mark it read-only, date its name, and stop updating it forever. Two systems both accepting edits is the four-file problem returning in a trench coat. — [The archive discipline](#)

Importing a spreadsheet, step by step

- 1 Tidy the sheet first**
One header row, one kind of thing per row, no merged cells, no notes floating in the margins. Fifteen minutes here saves an hour of repair afterward.
- 2 Export to CSV**
From Excel or Google Sheets, save a copy as CSV. Notion reads that cleanly where it may stumble on complex spreadsheet formatting.
- 3 Import into the database**
Use Merge with CSV on an existing database to add rows to it, or import as a new database to start fresh. Notion maps columns to properties; unmatched columns arrive as new properties.
- 4 Fix types, then relate**
Set each property to its real type — date, select, email — before linking anything. Then make the relations, which is the pass that turns a list into a system.

A backup rhythm worth keeping

- export the workspace from settings and store the file where your other backups live. Ten minutes, four times a year, is enough for most offices; monthly is better once real money data is inside. — [Monthly](#)
- export ahead of a bulk import, a schema change, or handing access to someone new. The undo you never need costs nothing. — [Before anything big](#)
- page history rewinds edits, and Trash holds deleted pages until you empty it. Most disasters are ten seconds of recovery, not a support ticket. — [Know the two safety nets](#)

The human layer that makes it stick.

- every workspace needs a person who holds the keys and the tidying habit — even in a volunteer-run shop. Owners set permissions, adopt or remove stray databases, and answer the where-does-this-go questions in week one so nobody invents a second filing system. — One named owner
- dated meeting notes (2026-03 Board), a ■ prefix or dedicated section for source databases, verbs for buttons. Twenty minutes of convention saves a year of hunting. — Naming conventions, small and early
- agendas live on the meeting's page before it starts; notes are taken on it during; decisions and owners get bolded; the page is linked from the project it affects. One meeting run this way teaches the whole pattern. — Meeting hygiene
- onboarding that says here's where your job lives beats a features tour every time. Each staff member's first session should end with them entering something true — a family, a gift, next month's program. — Train on real work
- a one-page If you're new here — where things live, how we name, who owns what — is knowledge management in its smallest viable form, and the page future-you will be most grateful for. — Write the culture down

Naming conventions that pay off

KIND OF THING	NAME IT LIKE	SO THAT
Meeting notes	2026-03 Board Meeting	They sort chronologically without a date property
Source databases	■ Families, ■ Donations	Sources are visually distinct from the pages that view them
Buttons	Log Touchpoint, Create Invoice	A verb tells a volunteer what will happen when they click
Views	All People, Expiring Soon, Active Care	The name states the filter, so nobody has to open it to know
Archived files	ARCHIVE 2025 Membership (read-only)	Nobody edits the retired copy by accident
Policy pages	How we handle Do Not Contact	Search finds them by the question people actually ask

Two documents worth writing on day one

- one page: where things live, who owns each area, how we name things, and the three habits we keep. It is the cheapest insurance against key-person risk that exists. — If you're new here

- 
- **equally important — the boundaries around what belongs in secure systems elsewhere, written down before someone has to guess.** — What we do not put in Notion

Thirty days to a working desk, then sixty to a habit.

1

Week 1 — Foundation

Create the workspace or install a template, apply for the TechSoup discount, set the areas, lock the sensitive doors, name the owner, and enter ten real households.

2

Week 2 — The directory lives

Finish households and people, set contact preferences, and retire the old membership spreadsheet to a dated, read-only archive.

3

Week 3 — Money and calendar

Start the giving log with this month's gifts linked to their givers; put next month's programs on the shared calendar; invite the treasurer and program staff to their areas.

4

Week 4 — Rhythm and review

Run the first weekly fifteen-minute review, assemble the month's numbers for the board from the workspace, and choose the next area — only the next one.

Days 30 through 90 are where the habit sets: add one automation the office already performs manually, run one form for a real event, bring the board hub live, and backfill history only in quiet weeks. The test of success stays a single question — when someone in the office needs an answer, do they open the workspace first?

Six pitfalls, named so you can dodge them

One giant table instead of related databases. Copies instead of views. A workspace nobody owns. Launching eight systems at once. Demo or test data left to rot among real records. Permissions treated as an afterthought. Every failed first attempt this guide has seen traces to one of the six — and every one is a pattern choice, avoidable on day one for free.

The rollout at a glance

BY THE END OF	YOU SHOULD HAVE	SIGNAL IT WORKED
Week 1	The workspace, permissions set, an owner named, ten households entered	Someone asks a question and opens Notion first
Week 2	The full directory, contact preferences set, old sheet archived	The office stops opening the spreadsheet
Week 3	This month's giving logged, next month's programs on the calendar	The treasurer checks a total without asking anyone
Week 4	The weekly review run once, the month's board numbers pulled	The board report takes minutes instead of an evening
Day 90	A second area live, one automation running, a form collecting sign-ups	New staff are onboarded by reading, not by shadowing

The short answers.

- **Notion encrypts data in transit and at rest and runs on major cloud infrastructure; business and enterprise tiers add SSO and audit controls. The larger risk in small organizations is permissions set casually — which chapter 10 exists to prevent.** — Is Notion secure enough for member data?
- **the volunteer who can use a document and a table can use Notion; the design burden falls on whoever builds the views. Give each role one dashboard with one obvious place to type, and adoption follows.** — Can volunteers with no tech background use it?
- **if the patterns in this guide were followed, they leave behind structure instead of secrets: named databases, legible views, a written if-you're-new page. That's the difference between knowledge management and a key person risk.** — What happens when the one Notion person leaves?
- **no — the relational core delivers most of the value. The AI compounds it for the one or two people doing the writing, searching, and cleanup; buy those seats when the need is felt, at the Business tier.** — Do we need the AI?
- **dedicated platforms are capable and cost \$1,600–\$2,400 every year at entry tiers. Notion trades turnkey modules for ownership, flexibility, and a fraction of the cost — and a purpose-built template closes most of the assembly gap in an afternoon.** — Notion versus dedicated synagogue software?
- **the member directory, this week, with ten real households. Every other decision gets easier once the office trusts one shared list.** — Where should we start, honestly?
- **yes — edits appear live, with each person's cursor visible. The merge conflicts that plague shared documents simply do not occur here.** — Can two people edit the same page at once?
- **recently opened pages are available in the desktop and mobile apps, and edits sync when the connection returns. Treat it as an online tool that tolerates a tunnel, not an offline-first one.** — Does it work offline?
- **an afternoon to be useful, a month to be fluent, and the fluency compounds. Chapter 1's four-move exercise covers most of what a staff member needs on day one.** — How long does it take to learn?
- **attach them to the records they belong to — the signed contract on the facility booking, the flyer on the event. Files stop living in a downloads folder and start living in context.** — What about our old files and PDFs?
- **the mobile apps handle looking things up, logging a touchpoint, and checking a schedule comfortably. Building views and databases is a desk job.** — Can we use it on phones?

The fourteen words, in one place.

TERM	MEANING
Block	The atomic unit of content — paragraph, heading, image, table — movable and transformable
Page	A block that holds other blocks, including other pages; the unit of sharing
Teamspace	A top-level section with its own membership — one per functional area
Database	A collection of records; the structured heart of the workspace
Record	One row of a database — and simultaneously a full page of its own
Property	A field attached to every record: select, date, number, person, and the rest
View	One presentation of a database — table, board, calendar, gallery, chart — with its own filters
Linked view	A window into a database placed on another page; what dashboards are made of
Relation	A live two-way link between records in different databases
Rollup	A property that computes across a relation — sums, counts, latest dates
Formula	A property that computes within a record from its own fields
Template (database)	A pre-built page for new records; can repeat on a schedule
Automation	A trigger-and-action rule a database runs on itself
Guest	Someone granted specific pages without joining the workspace

Where this guide comes from

It was written by Jacob Sager of Brand New Colors, who does knowledge management and AI builds for real people in the real world — and who built Shalom Desk, a complete Notion workspace for Jewish community organizations with thirty connected databases across eight areas, by sitting with a working community and encoding the shape of its week. The full workspace is browsable as a live read-only preview at [YOUR-SITE-URL]. This guide stands on its own either way — take the patterns and run.