Sample report on our own demo app

SAMPLE REPORT ON OUR OWN DEMO APP · MADE-UP DEMO DATA

PatchAndShip

Sample report on our own demo app · PatchAndShip Database Check ($49 process) Order: SAMPLE (no customer, no payment) · Scope: our own demo Supabase project `patchandship-demo` (ref `hvfgyjyytmixpvczcfoz`) Script: our customer read-only script v1-2026-10-18, run as `postgres` · Before run: Wed Oct 28 2026 11:37 PM PT · After run: Wed Oct 28 2026 11:37 PM PT Read-only for the check itself: the script reads settings and rules, not table rows. (Unlike a customer order, we then fixed our own demo project to show the "after".)

> Everything here is made-up demo data (demo_user_1@example.com, "Demo note A", a text file saying "Demo file. Not real data."). We planted the mistakes on purpose. No real people, no customer.
>
> Note on grants: after Supabase's Oct 30 2026 change, NEW tables get no grants to anon/authenticated by default. We granted them explicitly in the seed so the demo looks like older projects and common tutorials.

1. Summary

Before: **4 red, 2 yellow, 1 green.** Two tables could be read or changed by anyone holding the public key, an upload bucket was public and took anonymous uploads, and an older demo table lets anyone read it. After the fixes: **1 red, 2 yellow, 4 green.** The three planted mistakes are fixed. The remaining red and yellows are on older demo objects (`demo_orders`, `demo_customers`, `demo-public`) we left unchanged on purpose because other demos use them.

2. Fix these first (BEFORE, in order)

#	Rating	Area	Finding	Evidence	What to do
1	Red	RLS per table + grants	`sample_notes` has Row Level Security OFF, and anon/authenticated have read and write grants. In plain words: anyone with the public key can read, edit or delete every note. Security Advisor lists it as ERROR "RLS Disabled in Public".	before-script L03, L11, L12; before-advisors `rls_disabled_in_public`	Turn RLS on, add per-user policies, remove anon grants.
2	Red	Policies	`sample_profiles` has RLS on, but its policies say `true`: logged-out visitors can read every profile (incl. emails), and any logged-in user can edit anyone's profile. The Security Advisor did not list this one; our script's policy check did.	before-script L06, L07, L13, L14	Replace `true` with `(select auth.uid()) = user_id`.
3	Red	Storage	Bucket `sample-uploads` is public and a policy lets logged-out visitors upload into it with no folder/owner limit. Anything uploaded can be fetched by anyone who has the link.	before-script L16, L08	Make it private; let users read/upload only in their own folder.
4	Red	Policies + grants (older demo table, left unchanged)	`demo_orders`: policy "demo: anyone can read orders" is `true` for anon, and anon has write grants (RLS still blocks writes because there's no write policy).	before-script L05, L09, L10	Scope the read policy to the owner; revoke anon writes.
5	Yellow	Storage (older demo bucket, left unchanged)	Bucket `demo-public` is public. Fine for public images; not fine for private files.	before-script L15	Confirm it only holds public assets.
6	Yellow	RLS	`demo_customers` has RLS on but no policies, so the app can't read it with the public key at all. Not a leak; may be a broken feature. Advisor: INFO "RLS Enabled No Policy".	before-script L01; before-advisors `rls_enabled_no_policy`	Add owner policies if the app needs it.

3. Green / no issue found (BEFORE)

- No SECURITY DEFINER functions and no views without `security_invoker` in user schemas (script sections 5 and 6 returned no rows; before-script has no such lines).

4. The fixes we applied (fix.sql)

- `sample_notes`: RLS on; 4 policies (select/insert/update/delete) that only match rows where `(select auth.uid()) = user_id`; all anon grants revoked.
- `sample_profiles`: the two `true` policies dropped; select/update policies scoped to `(select auth.uid()) = user_id`; anon grants revoked.
- `sample-uploads`: bucket set to private; the anonymous upload policy removed; logged-in users can read and upload only in a folder named after their user id.

5. AFTER (same script, same Advisor)

Rating	Item	Evidence
Green	`sample_notes`: RLS on, 4 per-user policies, no anon grants. Advisor no longer lists it.	after-script L03, L06–L09, L16; after-advisors
Green	`sample_profiles`: per-user select/update only, no anon grants.	after-script L04, L10, L11, L17
Green	`sample-uploads`: private, own-folder policies only.	after-script L19, L12, L13
Green	No SECURITY DEFINER functions / views without security_invoker.	after-script (no section 5/6 lines)
Red	`demo_orders` (unchanged older demo table): same as before.	after-script L05, L14, L15
Yellow	`demo-public` (unchanged): public bucket.	after-script L18
Yellow	`demo_customers` (unchanged): RLS on, no policies. Advisor INFO still listed.	after-script L01; after-advisors

Logged-in users still hold table grants on `sample_notes` / `sample_profiles` (after-script L16, L17); the per-user policies limit those to their own rows, and actions with no policy (e.g. deleting a profile) are refused by RLS.

6. Not covered in this sample

Key scan of a live app bundle and repo: the demo project has no app front end in scope here, so we did not run it. No live REST tests with the public key (those are outside the $49 Check).

7. Disclaimer

Point-in-time review of what the script and Security Advisor showed. Not a penetration test, audit, compliance certification or guarantee of security (see Terms and Refunds on patchandship.com).

Appendix: raw evidence (keys masked)

No keys appear in any output. Publishable key, if referenced: `sb_publishable_…GbXD`.

- `raw/before-script.txt`, `raw/after-script.txt` (numbered lines `Lnn` cited above; raw JSON in `*-raw.txt`)
- `raw/before-advisors-raw.txt`, `raw/after-advisors-raw.txt`
- `seed.sql`, `fix.sql`

Security Advisor

Before: ERROR rls_disabled_in_public (public.sample_notes); INFO rls_enabled_no_policy (public.demo_customers). After: INFO rls_enabled_no_policy (public.demo_customers) only.

Not affiliated with Supabase. Raw script output: raw/before-script.txt, raw/after-script.txt (full listing in the source report appendix).