

Create the app database

Define persistent request records and their ownership.

THE FINISHED RESULT

Valid records persist and the second account cannot read or change the first account's requests.

Before you start

Environment: React + JavaScript + Supabase for the connected path; plain JavaScript local demo

Access and plan: The supplied local demo needs only a browser. The connected path requires Node, a React project, a Supabase development project and approved test accounts.

Practice project: Harbor request board

Use a separate demo project and fictional information. Record the exact project and version you use. Keep passwords, verification codes and private customer information out of screenshots and notes.

Official walkthroughs

Supabase: [React quickstart](#)

Supabase: [Row Level Security](#)

Open the credited publisher walkthrough for current interface instructions and available screenshots, then return to the SETLY exercise. External courses remain with their publishers; this PDF does not reproduce their complete tutorial.

Your practice task

Finish one action at a time. Inspect the resulting screen or record before moving on. If a control differs, check the matching official instructions instead of guessing.

01 / Step 1

Create a development project with the chosen database provider. Define requests with `id`, `owner_id`, `title`, `service`, `status` and `created_at`.

Check: record what changed and confirm you are still working in the intended demo project.

02 / Step 2

Make required fields non-null and constrain status to the allowed values. Use server-generated identifiers and creation timestamps.

Check: record what changed and confirm you are still working in the intended demo project.

03 / Step 3

Enable row-level access controls and specify that signed-in users may access only records whose `owner_id` matches their identity.

Check: record what changed and confirm you are still working in the intended demo project.

04 / Step 4

Insert a demo row through the authenticated client and read it back. Test with a second account to check isolation, using the provider's documented workflow.

Check: record what changed and confirm you are still working in the intended demo project.

Test, troubleshoot, repeat.

If something goes wrong

If all users can read all rows, fix database policies before adding interface filters. Hiding rows in JavaScript is not authorization.

If your screen is different

Compare the product, editor, user permissions and selected account with the guide. Open the official reference for your version. Do not perform a live change whose effect you cannot verify.

Final verification

Valid records persist and the second account cannot read or change the first account's requests.

Repeat the task with one small variation. Record the test date, project/version, expected result, observed result and a non-sensitive evidence reference. Mark skipped checks as untested.

Your evidence record

Test date: _____ Project/version: _____

Observed result: _____

Evidence reference: _____

Remaining issue and next action: _____

Sources and guide status

Original exercise by Ray / SETLY. Credited technical references are linked on page one. Prepared 1 October 2026. SETLY screen recording and independent learner reproduction are not yet complete. No platform affiliation is implied.

[Open the online practice guide and editable checklist](#)