AURUM ACADEMY · SAMPLE MODULE

FSP 203

Supabase I — PostgreSQL, Auth & Row Level Security

From the curriculum: Modern Full-Stack Product Engineering

Level	Duration	Credit units	Standalone
200 — Core Stack	5 weeks	5	NGN 65,000

Developed by the team behind 10 live products.

ABOUT THIS SAMPLE

Read before you buy. That's the point.

This is the complete outline for one of the eighteen modules in the Modern Full-Stack Product Engineering curriculum — exactly as it appears in the full programme document. We publish it so you can judge the depth for yourself before spending a naira.

We chose FSP 203 deliberately. The database and identity layer is where most self-taught developers have the deepest gaps, and where most tutorials are weakest. It is also the module our own products lean on hardest: every application we operate enforces authorisation with Row Level Security, audited to zero errors.

MODULE DESCRIPTION

The database and identity core of every product: relational modelling in PostgreSQL, SQL fluency, Supabase Auth, and security enforced at the database layer with Row Level Security. By the end of this module, the database itself is your authorisation system — not a hope that your application code never makes a mistake.

LEARNING OUTCOMES

On completing this module, you will be able to:

- Design normalised relational schemas: tables, keys, constraints, indexes, and relationships that survive real product growth.
- Write production SQL: joins, aggregations, views, functions, and migrations managed as code.
- Implement sign-up, login, email confirmation, and session management with Supabase Auth — including the redirect URL configuration that silently breaks most first deployments.
- Enforce authorisation with Row Level Security policies on every table — deny by default, allow by policy.
- Model role-based access (owner / manager / viewer patterns) at the database level.
- Audit a project with security advisor tooling and remediate findings: missing RLS, exposed SECURITY DEFINER functions, weak password protection.

WEEKLY BREAKDOWN

Five weeks. Each one ends in working, tested output.

Week 1 — Relational modelling

- Entities, relationships, and constraints: thinking in tables before writing any code.
- Primary keys, foreign keys, unique constraints, and when to index.
- The Supabase dashboard and SQL editor as a professional workflow, not a toy.
- Lab: model a multi-tenant application schema from a written product brief.

Week 2 — SQL in depth

- CRUD beyond the basics: upserts, returning clauses, transactions.
- Joins and aggregations that answer real product questions.
- Views and database functions; migrations written and versioned as code.
- Lab: a reporting query set over the Week 1 schema, reviewed line by line.

Week 3 — Supabase Auth

- Email/password flows, sessions, and JWTs — what actually happens on sign-in.
- Email confirmation and the auth redirect URLs that must match your deployment.
- The auth schema: what Supabase manages for you and what it expects from you.
- Lab: full sign-up → confirm → login → session flow on a deployed URL.

Week 4 — Row Level Security

- The deny-by-default mindset: RLS as the only authorisation layer you trust.
- Policies for select, insert, update, and delete; auth.uid() in practice.
- Role-based access patterns: owner, manager, viewer — enforced in the database.
- Lab: write and breach-test the complete policy set for the module project.

Week 5 — Security auditing

- Running the security advisor and reading its findings like an engineer.
- SECURITY DEFINER functions: when they are a risk and when they are acceptable.
- Leaked-password protection, and the remediation workflow to zero errors.
- Lab: a written audit report on your own project — the same artefact you would hand a client.

PRACTICAL PROJECT

The graduation gate for this module

Develop a multi-tenant data application in which Row Level Security is the only authorisation layer — no permission checks in application code. The project is verified by attempted breach testing: if another student can read or write data they should not, the project is not done.

ASSESSMENT

Component	Weight
SQL practical exam	30%
RLS project (with breach test)	50%
Security audit report	20%

PREREQUISITES

FSP 104 (React Fundamentals) or equivalent experience. Level 100 exists for complete beginners; experienced developers enter the programme directly at Level 200.

WHERE THIS MODULE SITS

FSP 203 is the third of six modules in Level 200 — The Core Production Stack — alongside Next.js 15, UI engineering, storage & scheduled jobs, transactional email, and deployment on Vercel. The full curriculum runs four levels and eighteen modules, closing with a guided capstone in which every graduate ships a live, secured, revenue-ready SaaS, verified by an automated audit suite and an AI-conducted viva.

“Every module ends in a shipped, verifiable artefact — because employers and customers pay for products, not certificates alone.”

TAKE THE NEXT STEP

- **This module, standalone:** NGN 65,000 — enrolment opens with the founding cohort.
- **The Core Stack Bundle:** all six Level 200 modules for NGN 195,000.
- **The Full Diploma — founding cohort price:** all eighteen modules and a guided capstone with automated verification and AI viva — NGN 495,000, or three instalments of NGN 175,000. Rises to NGN 750,000 at public launch.

Join the founding cohort at academy.aurumdigitalconsulting.com