

SAHIL SAPRA

Charlotte, NC 28273 | (216) 848-7236 | sahilsapra391@gmail.com | sahilsapra.com | [LinkedIn](#) | [GitHub](#)

PRODUCT MANAGER & DATA ANALYST | REVENUE ANALYTICS & DATA MODELING

August 20, 2026

[Affirm Hiring Team](#)

I am applying for the Analyst II, Full Stack role on the Revenue Analytics team. The line in your posting about owning end-to-end work across data modeling, metric definitions, dashboards, automation and enablement is the job I have actually done, starting from the analyst end of it and ending up owning the product around it.

My first analytics work was seven Tableau dashboards built on Snowflake SQL data models across four customer-facing applications. They were not decoration. They surfaced the usage patterns behind churn, and the changes that came out of them cut customer attrition 28%. That was where I learned the unglamorous half of this job: a dashboard nobody trusts is worse than no dashboard, so the model underneath it has to be right before anyone sees a number.

The piece of my work closest to revenue analytics started as an escalation. Million Air, our second-largest account, has 25 airport locations, and the CEO was receiving 25 separate weekly revenue emails from his FBO managers. Engineering had no bandwidth, so I built it myself: one dynamic PowerBI digest with cumulative revenue and per-location drill-down, delivered automatically every Sunday. Before any of it reached a CEO inbox I reconciled every figure against raw Snowflake data, because being wrong once at that altitude ends the product. Twenty-five messy emails became one verified digest at zero engineering cost, and an escalation turned into a stronger relationship.

Turning a vague problem into a system is the habit I would bring. Onboarding one enterprise customer onto our next-generation platform took over two weeks of manual ETL, and I owned a 75-customer migration program, which run sequentially is roughly three years of work. Instead of asking for headcount I broke the pipeline into its actual steps and automated the data injection ETL, taking capacity to three customers per day. All 75 moved with zero operational downtime, and ARR grew 104% year over year two years running.

On the AI half of the posting, I build rather than review. I founded SpecHawk, a deploy-aware no-code agentic QA platform, and grounded its test case generation in retrieval-augmented generation over roughly 2,000 of our own ticket-to-test conversions, so output followed house conventions and improved with each added example instead of requiring retraining. That work is the same argument your posting makes about data foundations: the model is only as reliable as the context and definitions you feed it. I use Cursor, Claude Code and Copilot in daily development, and I integrated AI tooling into our delivery workflow to pull a module redesign in by five months.

Where I am thinner is your specific stack. My modeling and SQL work has been in Snowflake with Tableau and PowerBI on top, not dbt, Databricks or Sigma, and I have not worked in Salesforce. I am comfortable saying that plainly because the transferable part is the reasoning about grain, definitions and trust in a number, and the tooling I would pick up quickly on the job.

Sincerely,
[Sahil Sapra](#)