Intro

Throughout my career, I've had three core goals:

- Never settle for "just works" when something could work better.
- Stay ahead of the curve with tools and techniques—often before they become industry buzzwords.
- Always bring more value than expected, especially by building systems that reduce the need for my ongoing involvement.

Before I transitioned into data engineering and analytics, I spent several years in inventory planning, purchasing, and restaurant management. While not directly tied to my later technical work, this background gave me a unique edge: I've been the consumer of data, I understand the pain points of stakeholders, I speak the language of non-technical users, and I know how to frame insights in terms of real business impact.

From the start of my data journey, I've pushed for continuous improvement—whether in efficiency, usability, engagement, or accuracy. I also bring a unique cognitive lens: I excel at pattern matching, anomaly detection, and structured thinking. I prefer raw data over flashy GUIs and take pride in delivering clarity over clutter.

Musician's Friend

Inventory Planning

I began my data career at Musician's Friend, a major omni-channel, multi-brand e-commerce retailer, as a Report Coordinator for the Inventory Planning department. My initial job was to manually manipulate flat files from a legacy ERP system in Excel and distribute them to planners—via print and email. It was an 8-hour grind—often utilizing two desktops, frying both a CPU and a hard drive—and critical daily reports sometimes weren't ready until 3 PM.

Step one: I automated Excel processing using VBA, maintaining the team's preferred layout while delivering reports by 10 AM.

Step two: I reduced ad-hoc requests by building a self-service data retrieval tool in Excel, using VBA and ODBC to pull filtered data from our DB2-based inventory system.

These improvements freed up half my day, giving me room to add more value. I briefly shifted into an Assistant Inventory Planner role but found I was still spending most of my time fulfilling reporting and analysis needs. Recognizing where I was most impactful, leadership promoted me to a Programmer/Analyst role.

In this role, I attended planning meetings, listened intently, and afterward built solutions to address the business problems I'd heard. I proved myself and was granted access to our Oracle data store and SQL Server Report Server and proceeded to rebuild the entire reporting suite. I created both self-service and scheduled reports, effectively eliminating the need for my original position.

My initiative earned me an Innovator Award for translating ambiguous business needs into actionable insights—even when stakeholders couldn't fully articulate what they needed.

At this time, the company began a major migration from its legacy ERP to Microsoft Dynamics AX, requiring the transfer of 25 years of sales and procurement data. The effort also included building an operational data store, a new data warehouse, and an SSAS cube. I trained the inventory and merchandising teams in Excel-based self-service no-code reporting using the new cube. Though I was still officially in Inventory Planning, I was fully embedded in the Data Management and Decision Support team, rewriting every relevant report for go-live.

Data Management and Decision Support

As the ERP transition progressed, I was promoted to Decision Support Analyst and moved formally to the Data Management and Decision Support team. This move was pivotal in my career, allowing me to work alongside seasoned developers who mentored me extensively.

I built hundreds of validation scripts to ensure 100% accuracy in the data migration to Dynamics AX. I also remained the go-to for inventory and sales reporting, though now with a broader scope.

When the transition was completed, I began developing ETL processes using SSIS to pull data from our warehouse management and inventory systems into our operational data store, replacing linked servers and reducing strain on production systems.

Eventually, my position was eliminated during a Bain Capital restructuring. I declined to relocate, gracefully trained my replacement, and departed with dignity during the company's final days in Southern Oregon.

Maintenance Connection

Technical Solutions Engineer

Next, I joined a small, distributed CMMS software company in Sacramento, CA. I was initially hired to take over data imports for a major client. The process was already automated, and I quickly informed my manager that it only took about 15 minutes a day. That task was reassigned, and I pivoted to making myself useful elsewhere.

I dug into the product's architecture and quickly became the go-to resource for database issues. I provided Level 2 support and managed installations for on-premises clients.

Database Administrator and Lead SQL Developer

It stunned me that the company had no dedicated DBA, despite over 1,000 tenants across eight SQL Servers in their SaaS environment. Backups were basic, there was no monitoring or integrity checks, and SQL Server was left in its default configuration for years. I convinced the founder that the days for the reactive, pray-it-doesn't-break startup mentality had expired, the role was formalized, and I was promoted to be their first DBA.

I began by resolving long-ignored issues:

- **Deadlocks:** I traced persistent deadlocks to recursive triggers with nested cursors. I resolved the issues and implemented a hotfix that virtually eliminated deadlocks.
- **Encryption:** The system stored user passwords using easily reversible obfuscation. I replaced it with a salted and peppered encryption method using per-user secrets, implemented secure triggers and procedures, and passed Apple's data security review during their adoption of our product.
- **Performance:** I resolved severe CXPACKET and LATCH waits by adjusting MAXDOP settings for OLTP best practices. Query performance and concurrency improved dramatically.
- **Backups:** By staggering backups across servers, I reduced I/O contention on the shared NAS, reduced network contention, and improved backup times by 25%.

I also spearheaded a hardware upgrade and server consolidation project, proposing a new server with modern specs that could replace three aging ones. After approval, I built a reusable scripted database migration process that successfully moved over 500 databases in under an hour, with service interruptions averaging less than 10 seconds per tenant.

Once the DBA role was stabilized, I was able to focus more on development as the Lead SQL Developer.

Core Product Upgrade

This major upgrade introduced multilingual support and cross-browser compatibility, but I also saw an opportunity to modernize the database layer.

I pushed to upgrade compatibility from SQL 2000 to 2008, unlocking CTEs, MERGE, TRY/CATCH, and more. Resistance came from concerns about legacy client support, but I demonstrated that:

- Microsoft no longer supported or provided security patches for SQL 2000.
- Clients using SQL 2000 were typically on SQL Express and were already unsupported per company policy.
- Recursive CTEs could drastically outperform cursors.

I rewrote the hierarchical retrievals using recursive CTEs, which resulted in a 1000-fold improvement from a few seconds to a few milliseconds. However, there was strong pushback from the original developer, who claimed my solution would lead to deadlocks. To prove the viability of my solution, I ran stress tests with 100+ concurrent executions, using the exact conditions that were supposed to trigger deadlocks. It was no surprise to me when it resulted in zero deadlocks. The CEO signed off. Ironically, the developer who opposed the change was the same one who wrote both the recursive triggers that did lead to deadlocks and the ineffective encryption.

I reviewed and modernized ~600 stored procedures and triggers, eliminating cursors where possible and improving performance across the board.

I also identified and fixed the root cause of long-standing app crashes caused by malformed custom SQL rules. I added TRY/CATCH blocks and error logging to the procedure, restoring usability and stability.

Mobile App and API

As Lead SQL Developer, I reviewed all SQL for both products and mentored developers on optimization and best practices. I followed the example of my mentors—empowerment without critique or condescension.

A standout win: a list pagination procedure for the mobile app was taking 30 seconds per page—and was going to be duplicated 350+ times. I built a dynamic SQL procedure using layered CTEs. It reduced query times to under 20ms and served all modules from one reusable object. For more details on this project, view it in my portfolio under "Mobile Application List Pagination Procedure."

The Lean Times

After nearly a decade in the field, I had to take an extended and unexpected break. I knew I'd return, so I stayed engaged by developing both soft and technical skills.

Educator

I taught GED classes—initially all subjects, later focusing on math. I developed individualized plans for students with TBI, autism/ADHD, dyslexia, and more. I earned Advanced Tutor Certification from the College Reading and Learning Association and built my communication, empathy, and problem-solving skills.

I kept my tech skills sharp with side projects: building educational tools, tracking systems in Excel, games, and tools.

Preparing to Reenter the World Where I Belong

Once back on track, I dove back into SQL and picked up Python, Power BI, and even dbt by:

- Building ETL pipelines and basic analyses in Python.
- Developing applications in Python, including a document storage app (detailed in my portfolio under "Document Storage Application").
- Learned Power BI and built interactive dashboards and paginated reports on day one.

The Next Chapter

Where I go from here is still being written—but I'm back in my element, equipped with a broad, deep toolkit and ready for the next challenge.