**Milestone Four – Enhancement Three: Databases**

For this milestone, I selected my database artifact, originally created in DAD 220. The original project focused on customer returns, but it had a basic structure and limited functionality — really just enough to meet the requirements at the time. It lacked full normalization, indexing, and more advanced query capabilities. I wanted to take this artifact to the next level, not just polish it up, but rebuild it to better demonstrate real-world database design and querying skills for my ePortfolio.

For the enhancement, I started fresh using DB Browser for SQLite, an open-source tool that lets you easily create and manage databases. First, I designed a brand-new database schema from scratch, focusing on normalization and proper relationships. I created three tables — *Customers*, *Products*, and *Returns* — ensuring that each table had a clear primary key and that the relationships between tables were enforced through foreign keys. I made sure to set appropriate data types and add NOT NULL constraints where needed to promote data integrity.

Once the tables were built, I loaded them with sample data to simulate real-world records. I then built indexes on the CustomerID and ProductID fields to improve query performance, which is an important practice when designing scalable databases. After setting up the schema and data, I moved on to writing SQL queries. I didn’t want to just run basic SELECT statements — I built more complex multi-table JOINs to pull return records along with customer and product information, added aggregation queries to calculate return counts by reason, and wrote sorting queries to identify top return reasons by frequency.

All queries were tested directly in the Execute SQL tab in DB Browser. I kept things efficient and clean by using aliases, logical ordering, and ensuring that each query produced meaningful, readable results. Once everything was finalized, I exported the entire database to a .db file and saved a .sql export script. To round things out, I documented the schema and took screenshots of the key queries and results to show off the functionality.

This enhanced database project strongly aligns with Course Outcome 4 (CO4) by demonstrating the use of innovative techniques like indexing and optimization within a database context. It also addresses Course Outcome 5 (CO5) by emphasizing database design that enforces relational integrity and improves security, ensuring no redundant or orphaned data. Additionally, the structure and clarity of my queries and design decisions hit on Course Outcome 2 (CO2), which focuses on professional-quality communication through documentation and structured data management.

Throughout this process, I learned a lot about how the little details in database design — like properly setting up keys and constraints — make a huge difference in performance and usability. One of the biggest challenges was balancing normalization with practical query writing: too much normalization can make JOINs complicated, but without it, the database becomes inefficient and harder to maintain. Working hands-on with SQLite gave me a better appreciation for building databases that are not only functional but efficient and ready to scale.

Overall, this enhanced version is a huge improvement over the original DAD 220 project. It’s clean, efficient, and demonstrates my skills in modern database design and SQL query development. It also reflects how I approach problem-solving: start simple, build smart, and always focus on making things clear and efficient. I’m excited to include this artifact in my ePortfolio as proof of my ability to design databases that work just as well behind the scenes as they do on the surface.