Database System Development and Implementation Plan

Dana Woodard

Colorado Technical University

CS660: Database Systems

Dr. Kris Jamsa

08/10/2025

Table of Contents

[Database System Overview 3](#_Toc205491491)

[Database System Goals and Objectives 4](#_Toc205491492)

[Addressing Business Problems with Database Systems 5](#_Toc205491493)

[System Configuration with Marc’s Goals 6](#_Toc205491494)

[Implementing the Planned Goals 7](#_Toc205491495)

[Week Two 8](#_Toc205491496)

[Designing a Practical Database for Retail Workflow 8](#_Toc205491497)

[Laying the Foundation with Real-World Entities 8](#_Toc205491498)

[Protecting Data with Business Rules That Make Sense 9](#_Toc205491499)

[Attributes and Data Types 9](#_Toc205491500)

[Visualizing the System 10](#_Toc205491501)

[WEEK THREE 16](#_Toc205491502)

[Database Definition Language (DDL) 17](#_Toc205491503)

[Suppliers Table 19](#_Toc205491504)

[References 23](#_Toc205491505)

# Database System Overview

Marc’s Health Food Store has become a trusted part of the community. They are known for being a place where people like to shop and connect with their community. The focus on organic, raw, and locally sourced products reflects a growing need to take better care of our bodies, our families, and the environment. With recent growth comes new responsibility. Customers have started asking for online shopping options, not just out of convenience, but because life demands flexibility.

A working mom trying to prep healthy meals after a long day or a customer recovering from surgery may not have the time or ability to visit in person. Currently, the store’s website is more like a static flyer than a digital storefront. There is no option to order products online, register for workshops, or interact with content. At the same time, the current in-store database is unreliable, making it difficult to track inventory, manage customer preferences, or generate reports that could help guide business decisions. That creates gaps not only in service but in the ability to grow and sustain relationships.

Marc’s financial partner emphasized the urgency of increasing revenue by at least five percent or risk closure. That is not just a business issue, it affects local farmers, loyal customers, and the employees who help keep things running. Investing in a strong and secure site will help them be able to build scalable database systems. This will be a huge technical upgrade. It is a practical way to serve better, operate with wisdom, and create space for healthy, sustainable growth rooted in care for people and community.

## Database System Goals and Objectives

The objective of the recommended database structure is to establish a reliable, integrated platform that supports the full scope of Marc’s Health Food Store’s operations, both in-store and online. One of the top priorities is improving inventory management. At present, there is no consistent way to track which items are in stock, which are selling quickly, or what needs to be reordered. A modern database system will provide real-time inventory tracking, helping the staff restock efficiently and avoid missed sales due to out-of-stock products (TRACTIAN, n.d.). Another major objective is enhancing customer relationship management. We will need to securely store profiles, shopping history, and preferences.

This data can be used to personalize service and communication, which helps increase customer retention and engagement (Salesforce, 2023). One example could be customers who frequently purchase gluten-free items could be offered recipes or invited to related workshops. The database will also power an updated website, allowing customers to place online orders, access printable products and health information, and register for events. These features are essential for families and individuals. If a customer is unable to frequently visit the store or simply do not like to this feature would be immensely convenient for them.

Looking at the back end, the structure will routinely create sales reports, reports on product trends, and workshop performance. This will allow Marc’s team the insights needed to make informed decisions. These types of insights are critical for small businesses trying to compete with larger, more automated competitors (Forbes Technology Council, 2021). Finally, the system will integrate with the current point-of-sale platform so that all transactions, both online and in-store, are recorded in one place. This creates a clear, accurate picture of how the business is performing day by day. Overall, the database system is not just a technical tool. The database system will be a foundation for smarter operations. This will include more personalized service and steady growth that is rooted in real customer needs.

## Addressing Business Problems with Database Systems

The database system will help solve some of the key business problems Marc’s Health Food Store is currently experiencing. These problems currently include unreliable inventory tracking, limited access to customer data, and a lack of online functionality. At the moment, staff members cannot easily determine what items are in stock or which products are selling quickly. This can lead to over-ordering, understocking, or missed opportunities to highlight popular items. A centralized database system will solve this by providing real-time inventory visibility and automated stock updates (TRACTIAN, n.d.). The system will also allow the store to collect and store customer information, such as previous purchases and product preferences. This kind of data is essential for building meaningful customer relationships and offering personalized service, such as tailored recommendations or reminders about upcoming workshops (Salesforce, 2023).

Another major issue is that the existing website does not permit clients to browse or shop the store online. Currently, customers cannot place orders, register for events, or download health-related resources. This is a very big disadvantage for a business that depends heavily on education and community engagement. This lack of digital functionality limits its reach. The new database system will support a dynamic website that will allow customers to view product availability, sign up for events, and access educational content.

This new website will also be able to automate key business processes. These processes include sales tracking, reporting, and customer communication. These improvements will reduce the workload on staff. The new website will also help improve the accuracy of business decisions by providing up-to-date, organized data. By addressing these problems through a single, integrated system, the store will operate more efficiently, serve its customers more effectively, and remain competitive in a growing digital marketplace (Forbes Technology Council, 2021).

## System Configuration with Marc’s Goals

The planned database system directly aligns with the mission and strategic goals of Marc’s Health Food Store. This will help by equipping the business with tools that make it easier to serve, grow, and connect with the local community. The store’s mission focuses on providing an all-inclusive shopping experience. Their target is aiming to reach individuals who want to maintain a healthy diet and are conscious or wanting to be consciously engaging in sustainable living. However, the current technology infrastructure limits access to resources and services, making it difficult to fully live out this purpose. By adding features like online ordering, printable health information, and automated workshop registration, the new database system will open up access to customers who are short on time, balancing busy schedules, or unable to visit in person.

This supports the goal of expanding the customer base by ten percent and delivering consistent education around health and sustainability. In addition, the system will allow the store to efficiently manage an expanding inventory of organic products, especially those sourced from local producers. This helps meet the strategic goal of expanding product category offerings while deepening partnerships with the local farming community.

The database will also allow the staff to track customer trends, measure workshop attendance, and evaluate which services are the most impactful tools that allow the team to grow thoughtfully, without straying from the store’s core values. According to the Harvard Business Review, small teams that adopt agile systems are better positioned to respond to customer needs in real time. These teams also are capable of staying aligned with their mission (Rigby, Sutherland, & Noble, 2018).

In today’s digital economy, customers expect convenience, personalization, and transparency when engaging with retailers. Shopify’s global retail report notes that consumers now want unified shopping experiences. They want these shopping experiences to connect physical and online spaces, especially when health, wellness, or education are involved (Shopify, 2022). The new database system supports this exact shift by combining internal business efficiency with external community engagement. As a result, it creates the kind of responsive, mission-centered structure that the health food store needs to thrive where there is a lot of competition but still values driven marketplace.

## Implementing the Planned Goals

The store has a mission that is community oriented. They aim to provide an all-inclusive feel for those pursuing healthy eating. They seek to help those who want a sustainable lifestyle and understand those types of practices. The store’s strategic goals support this mission through four measurable objectives. One is to increase the customer base by ten percent. Second, expand product category offerings. Third, its aim is to make product and health information easily accessible. Lastly, its aim is to increase sales by five percent (Shopify, 2022). The suggested database structure is meant to clearly support each of these goals. It is meant to help it be practical yet still measurable.

First, the system will help attract new customers by offering the convenience of online shopping, which is essential in today’s fast-paced environment. When customers can browse, learn, and order without stepping foot in the store, it removes barriers to entry and creates space for growth. Second, expanding product categories will be easier to manage because the database will organize inventory by supplier, category, and demand, giving Marc’s team the ability to quickly assess which items are performing well and which vendors offer the best partnership opportunities.

Third, the system will house all educational content. This content will include things such as recipes, product descriptions, and health articles, which can be printed on demand or accessed digitally. This helps ensure that customers are not only shopping, but learning, which supports long-term loyalty and deeper engagement. Finally, the system will produce sales reports that track performance against revenue goals, giving the team accurate data to evaluate success and make adjustments. According to Harvard Business Review, technology that improves access to real-time data is one of the most effective ways small businesses can stay aligned with their core mission while adapting to growth (Rigby, Sutherland, & Noble, 2018). If each of the organization’s goals are addressed in the platform the database system becomes a foundational tool that turns vision into action.

# Week Two

## Designing a Practical Database for Retail Workflow

When building a relational database for a retail store it is imperative to begin with a clear understanding of what the business needs are, be able to track those needs and how that information connects. A strong database is not just organized; it mirrors how the store operates day by day. The database must include identifying key subjects. These key subjects are things like outlining rules that protect the data and creating a visual model that shows how everything fits together. To keep the structure clean and reliable, the design must meet at least Third Normal Form. This helps reduce unnecessary repetition, improves accuracy, and supports future growth.

## Laying the Foundation with Real-World Entities

Every retail business revolves around a few core pieces of information. For this project, three essential entities are already provided: Customers, Orders, and Products. These reflect the natural flow of retail activity. Customers place orders, and those orders contain products. To make the database more complete, three more entities are added based on the needs of a real store. The first entity is Employees. This will allow the store to track which team member handled which order. This will help to pinpoint training and allow data to be collected that is accurate. Next is Categories, which group products into types such as “Shoes” or “Electronics.” Finally, the store must track its Suppliers, so it always knows where inventory is coming from. Together, these six entities create a balanced system that reflects the true structure of the business.

## Protecting Data with Business Rules That Make Sense

A database needs guardrails. These are called business rules, and they make sure that data is consistent, accurate, and tied to how things actually work. In this system, six business rules are used. The first rule says each product belongs to one category only. This helps staff and customers find items quickly and keeps reports clean. The second rule says a person becomes a customer only after placing an order. This avoids clutter from people who browse but never buy. The third rule gives each product a minimum reorder level. This triggers alerts when stocks are low, so the store does not run out of high-demand items.

The fourth rule assigns each order to one employee. This supports accountability, customer service follow-up, and performance tracking. The fifth rule lets a customer have more than one shipping address, but only one billing address can be the default. This keeps payment information consistent while allowing flexibility in delivery. The sixth and final rule says that products can only be linked to approved suppliers. This helps avoid errors, improves product quality tracking, and ensures the store can respond quickly to supply chain issues. These rules reflect real retail logic and keep the database useful and reliable.

## Attributes and Data Types

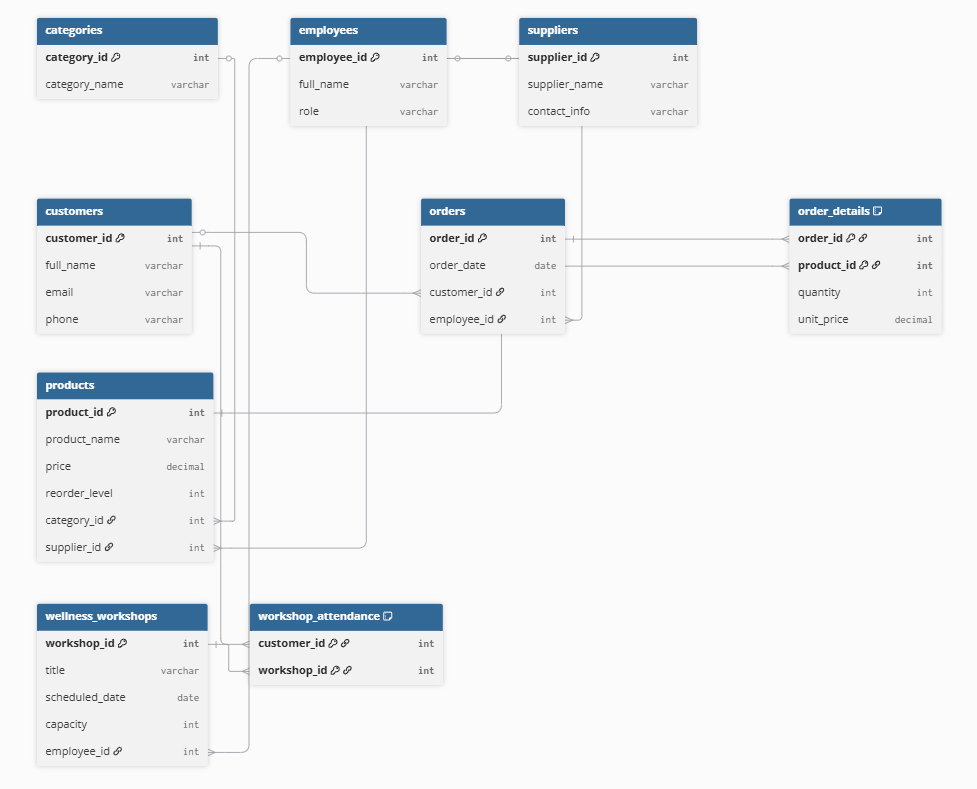
Each of the six entities has its own unique set of attributes. For example, a Customer includes fields like Full Name, Email, and Phone Number. An Order includes Order Date, Customer ID, and Employee ID. Products include Name, Price, Reorder Level, and links to both a Supplier and a Category. Employees have names and roles, while Suppliers include business names and contact information. Each attribute will use a suitable data type. The data types will be things such as integers for IDs, text for names, and decimals for prices. This keeps the database accurate and prevents format issues when staff are entering or retrieving data.

**Explaining How Everything Connects**

In addition to listing entities, the system must explain how they relate to each other. For example, one customer can place many orders, but each order belongs to one customer. One employee can handle many orders, but each order is linked to only one staff member. Each product is placed in one category, and one supplier can provide many products. Orders and products have a many-to-many relationship. To handle this, a junction table will be used, often called OrderDetails, which will include extra fields like quantity and unit price. These relationships are based on how things actually happen in the store and are essential to building an accurate model.

## Visualizing the System

To make the database easier to understand, an Entity–Relationship (ER) diagram has been created. This diagram shows each entity, the different attributes, and how it connects to the others. The ER diagram also includes keys and data types. Visual tools like this are helpful for developers and business leaders alike, as they turn abstract ideas into something concrete. Seeing the system laid out visually helps make sure everyone is on the same page before building begins.



**Keeping It Clean with Third Normal Form**

A strong database should be well organized, without repeated information or confusing overlap. That is why this system follows Third Normal Form. This means every field belongs in the correct table, and each piece of data is only stored once. For example, supplier details are only stored in the Supplier table, not repeated for every product. There are no circular or dependent relationships that could create errors. By creating a space that has a clean structure it will be simple to update information without breaking the system. This step is technical but necessary to make the database efficient, stable, and easy to maintain.

**Final Thoughts on the Retail Store Database**

This proposed database design gives the store a solid foundation to work with. It tracks the right information, follows realistic rules, and supports the kind of reporting and decision-making that store managers need. It also provides a logical and maintainable way to grow as the store expands or takes on more complexity. With the right structure in place the business will be better equipped. The new database will be able to handle orders and serve customers more effectively. It will be able to manage inventory effectively with the updated database.

**Visualizing the Store’s Data System: Entity–Relationship Model**

Designing a database is like building a blueprint for how the business will store and organize its information. An entity–relationship (ER) model helps map that out. This model is a visual planning tool that shows the key parts of the system, how those parts relate to each other, and what kind of information needs to be stored. It makes it easier to explain the structure of the database before any technical development begins (Harrington, 2016).

Each major subject the store needs to track is called an entity. An entity is a category of information, such as Customers, Products or Orders. Within each entity are attributes, which are the details that the business wants to keep about each item. For example, the Customer entity includes name, phone number, and dietary preference. The Product entity includes the product name, unit price, how many items are in stock, and the supplier data.

Each entity uses a primary key. The key, which is a unique code or number is utilized to track individual records. For example, every customer has a unique CustomerID, and every order has its own OrderID. These unique identifiers help the system keep information separate and accurate. When one table needs to reference another, such as when an order needs to identify which customer placed it, the system uses a foreign key. A foreign key is simply a copy of that unique ID that links back to the related table (Coronel & Morris, 2018).

The diagram also shows how the different entities are related to one another. For example, one customer can place many orders, but each order is tied to one customer. This type of connection is called a one-to-many relationship. Another relationship exists between products and orders. Each order may contain several products, and each product may appear in many different orders. This is called a many-to-many relationship and usually requires a separate table to store the extra details, such as quantity and price per product (IBM, 2023).

These relationships are displayed using crow’s foot notation, which is a visual method for showing how many records in one table relate to records in another. A symbol with three prongs represents “many,” and a single line represents “one.” This makes it easy to see at a glance how customers, orders, employees, and products are all connected (Lucidchart, 2024).

One special feature of Marc’s Health Food Store system is the addition of the Wellness Workshop entity. This table represents the community education programs the store offers. The store will offer things such as nutrition classes or food preparation demos. For each workshop it is led by an employee and can be attended by several customers. By adding this feature, the database design supports not just sales and inventory, but also community engagement and customer education, which are core parts of the store’s mission.

In simple terms, each table in the database tracks a different part of how the store operates. The Customer table stores basic contact details along with dietary preferences and any loyalty points the customer has earned. The Order table records what was purchased, who made the purchase, and the date it was placed. The Product table keeps track of the items the store sells. It also keeps track of their prices, current inventory levels, and which supplier provides each one. The Category table is used to group products into types such as produce, supplements, or household items.

The Supplier table holds important information about the vendors the store works with, such as their names and contact details. The Employee table stores staff names and job roles so that tasks such as order processing or workshop instruction can be assigned and tracked. Lastly, the Wellness Workshop table organizes information about customer education sessions, including class titles, scheduled dates, capacity limits, and which employee is leading each session.

This design helps all departments of the store, from the cashiers to the inventory team to the wellness instructors, work with the same system. It also helps the owner and managers see the big picture, from how products are selling to how many people attend classes. With this ER model in place, the database can be developed in a way that supports both business operations and customer care in a seamless, organized structure.

**Ensuring a Clean Structure with Third Normal Form**

Ensuring that a database is well organized begins with a process called normalization. Third Normal Form, or 3NF, is a largely accepted standard that helps prevent data duplication. This also helps the system to ensure that each table contains only the information it is meant to store. In a properly normalized system, every column in a table must depend directly on the table’s primary key, and there should be no indirect or transitive relationships between non-key fields. The proposed database design for Marc’s Health Food Store follows these guidelines.

For example, product details such as unit price, stock level, and reorder threshold are stored only in the Product table. The supplier information is stored separately in the Supplier table. By utilizing this structure, it helps to avoid repeating supplier details for every product. It also helps to create clean data and allows for cleaner updates. Similarly, customer contact details, loyalty points, and dietary preferences are kept in the Customer table and are not duplicated elsewhere. The system uses foreign keys to link related records across tables, which keeps the data connected without making it repetitive. The Order table includes both a CustomerID and an EmployeeID. These then can be linked back to the full records stored on their respective tables.

This design ensures that if a customer updates their phone number or a supplier changes their contact email, the system only needs to be updated in one place. While there are higher levels of normalization beyond 3NF, such as Boyce-Codd Normal Form, they are not necessary in this case because the current structure does not include multivalued dependencies or overly complex keys. For a small to mid-sized retail business that is expanding into many different areas such as online sales and education services, Third Normal Form provides a more balanced approach to the data system. The balance will bring the system clarity, flexibility, and be easier to use. This level of normalization will allow the system to run efficiently and ensure that the staff and management will have accurate data. It will be well-structured and easy to access.

**Supporting the Store’s Mission and Goals**

The proposed database solution directly supports the mission and core goals of Marc’s Health Food Store. It will support it by providing an organized, reliable, and scalable system that aligns with both daily retail operations and the store’s long-term purpose. As a health-centered retailer that values community wellness, customer education, and dependable service, the store requires more than a basic inventory tracking system. This database is designed to meet those needs by including features that reflect the food store’s goals of the organization. The inclusion of the Wellness Workshop entity allows the store to offer and manage educational events that promote nutrition, healthy living, and customer engagement. This feature helps the store fulfill its goal of creating a more informed and health-conscious community.

In addition, the business rules built into the database ensure that operations are accurate, consistent, and reflective of real-world expectations. These include rules for approved suppliers, employee accountability, and inventory restocking, all of which protect product quality and customer trust. The database also allows store leadership to track important metrics such as product movement. They will be able to see data on staff performance and things like workshop participation. This supports thoughtful planning and strategic growth. By organizing all of these functions into one cohesive system, the database advances the store’s mission to serve its customers with care, clarity, and a commitment to wellness. This will create for the food store a strong technical foundation. This will support both the business side and the community-centered goals of Marc’s Health Food Store.

# WEEK THREE

This stage of the project focuses on putting the database system into practice for Marc’s Health Food Store. The structured query language (SQL) will be used to develop and make the system’s architecture. This will help fill in the gaps with data, make updates, and generate reports. By utilizing these tools, it will provide both the technical structure and real-world usability needed to manage daily operations in a retail environment. The scripts developed in this phase allow the creation of tables for customers, products, orders, and more.

Once the structure is established, the system is populated with test data to simulate how the store will operate. There will also be data manipulation scripts that are used to update product pricing, remove outdated records, and add new customer information. In addition, reporting scripts generate insights such as monthly revenue per customer, product category performance, and workshop participation. The purpose of this project is not only to store information but also to help Marc’s team use that information to support decision making, customer care, and business growth.

From managing product reordering to tracking employee performance, the database plays a central role in the company’s mission to serve the community with high-quality food and wellness education. Each SQL script included in this report is explained in plain language, alongside technical terminology, so that both IT professionals and grocery store staff can become knowledgeable about how the system works and why it matters. This approach is in alignment with the principles of effective relational database design. That principle emphasizes clarity, normalization, and operational value (Coronel & Morris, 2018; Harrington, 2016). By following these best practices, the system becomes a useful and trustworthy foundation for day-to-day operations and long-term planning.

Database Definition Language (DDL)

The customer’s table is designed to hold information about each shopper. It tracks names, contact details, dietary preferences, and loyalty point balances. This will give Marc’s Health Food Store the ability to personalize service, support marketing campaigns, and segment customers for promotions or educational events. For example, the store could notify vegan customers about a plant-based workshop or offer loyalty rewards. By using a structured format for storing customer data, the business ensures information is consistent and easy to update when needed (Coronel & Morris, 2018; IBM, 2023).  
**See Appendix A (DDL.sql) for the complete Customers table script.**

## SQL Code: Customers Table See Appendix A (DDL.sql) for the complete Customers table script and sample inserts.

The customer’s table is designed to hold information about each shopper. It tracks names, contact details, dietary preferences, and loyalty point balances. This will give Marc’s Health Food Store the ability to personalize service, support marketing campaigns, and segment customers for promotions or educational events. For example, the store could notify vegan customers about a plant-based workshop or offer loyalty rewards. By using a structured format for storing customer data it will alllow the business to ensure that the proper information is consistent and easy to update when needed (Coronel & Morris, 2018; IBM, 2023).  
**See Appendix A (DDL.sql) for the complete Customers table script and sample inserts.**

SQL Code: Customers Table  
**See Appendix A (DDL.sql) for the complete Customers table script and sample inserts.**

Suppliers Table

The suppliers’ table stores information about the companies that provide Marc’s Health Food Store with inventory. This includes supplier names and email-based contact information so the store can place orders, follow up on deliveries, or address quality issues. Each supplier is assigned a unique ID so that products can be linked directly to their source. This structure helps ensure that all products in the system are traceable and can be tracked for reliability and consistency. It is important that you maintain a reliable supplier table. This table will be what supports inventory management, compliance, and supply chain accountability (Coronel & Morris, 2018; Lucidchart, 2024).  
**See Appendix A (DDL.sql) for the complete Suppliers table script and sample inserts.**

SQL Code: Suppliers  
**See Appendix A (DDL.sql) for the complete Suppliers table script and sample inserts.**

Categories Table

The categories table groups products into meaningful segments that store employees already use in day-to-day language, such as produce, beverages, and supplements. Categorizing items supports faster product lookup, cleaner shelf reporting, and accurate category sales analysis. It also helps with purchasing, since reorder decisions are often made by category. A clear category structure strengthens data integrity and makes reporting more useful for store managers and staff who rely on simple, readable groupings (Coronel & Morris, 2018; IBM, 2023).  
**See Appendix A (DDL.sql) for the complete Categories table script and sample inserts.**

SQL: Categories Table  
**See Appendix A (DDL.sql) for the complete Categories table script and sample inserts.**

Products Table

The products table lists every item the store sells. Each product includes a name, price, current on-hand quantity, and a reorder level that signals when to restock. The table also links every product to one category and one supplier so that staff can track sourcing and run category sales reports. This structure supports daily work at the register and in the stockroom, and it gives managers clean data for pricing reviews, margin checks, and purchase planning (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete Products table script and sample inserts.**

SQL: Products Table  
**See Appendix A (DDL.sql) for the complete Products table script and sample inserts.**

Orders Table

The orders table records every purchase made in the store. It stores the date of the order, the customer who placed it, and the employee who processed it. Because this part of the data links customer behavior with sales activity and staff performance, it makes it significant to the database. By having this data organized it will help ensure Marc’s Health Food Store is capable to stay on track. It will track purchase history, review seasonal sales trends, and follow up with customers when needed. The connection to the employees table helps management see who is handling the most orders, which is useful for scheduling and training decisions (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete Orders table script and sample inserts.**

SQL: Orders Table  
**See Appendix A (DDL.sql) for the complete Orders table script and sample inserts.**

Orders Details Table

The order details table records the specific items purchased on each order. It connects one order to one or more products, and it captures the quantity and the unit price at the moment of sale. This design keeps a clean history for reporting and audit purposes. When managers review revenue or margin trends, this table provides the line-item detail that explains which products are selling, in what quantities, and at what prices. It is the bridge between the orders table and the products table, which makes it essential for accurate revenue and inventory analysis (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete Order Details table script and sample inserts.**

SQL: Order Details  
**See Appendix A (DDL.sql) for the complete Order Details table script and sample inserts.**

Wellness Workshops

The wellness workshops table keeps track of the store’s community events. These events include things like nutrition classes and hands-on cooking demos. Each entry lists the workshop title, when it is scheduled, how many people can attend, and which employee is leading it. This helps the team stay organized, avoid overbooking, and plan for the right amount of space and materials. Because each class is tied to a specific staff member, it is easier to coordinate schedules and follow up with attendees. Offering workshops is part of Marc’s Health Food Store’s goal to go beyond selling products and create meaningful connections with customers by teaching them practical skills for healthier living (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete Wellness Workshops table script and sample inserts.**

SQL: Wellness Workshop  
**See Appendix A (DDL.sql) for the complete Wellness Workshops table script and sample inserts.**

Workshop Attendance

The workshop attendance table records which customers sign up for each workshop. It connects a customer to a specific event, showing their unique ID and the workshop they registered for. This makes it simple for the store to manage class rosters, check who has attended past sessions, and track repeat participation. With this information Marc’s Health Food Store can run their business more efficiently. It will allow them to follow up with attendees, send reminders about upcoming events, or suggest related products and classes. It also supports planning by showing which topics are the most popular, helping the team choose future workshop themes that match customer interests (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete Workshop Attendance table script and sample inserts.**

SQL: Workshop Attendance  
**See Appendix A (DDL.sql) for the complete Workshop Attendance table script and sample inserts.**

Employees Table

The employees table keeps a clear record of who works at Marc’s Health Food Store and what each person does. For every employee the table stores a unique ID, full name, and job role. This helps the team connect daily work to real people, such as linking an order to the cashier who processed it or a workshop to the instructor who conducted it. Managers can utilize this data to plan things like schedules, assign training, and follow up on service issues. Keeping these details in one place supports accountability and produces cleaner reports because names and roles stay consistent across orders and workshops (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete Employees table script and sample inserts.**

SQL: Employees Table  
**See Appendix A (DDL.sql) for the complete Employees table script and sample inserts.**

Data Manipulation Language (DML)

This section presents the core data manipulation statements that support daily work at Marc’s Health Food Store. The examples include removing an order safely, updating a product price, and three custom scripts that reflect common store tasks. Each script is written in a clear, repeatable form so staff and administrators can execute and verify results with confidence.  
**See Appendix B (DML.sql) for all DML scripts and verification queries.**

Deleting an Entire Order by Order ID

Removing an order requires deleting its line items first so that referential integrity is preserved. Using a transaction ensures that either both steps complete or neither step changes the database.  
**See Appendix B (DML.sql) for the complete script.**

Updating a Product Price by Product ID

Price updates are routine in retail. This statement changes the price for a single item identified by its unique product ID.  
**See Appendix B (DML.sql) for the complete script.**

Custom DML 1: Awarding Loyalty Points From an Order Total

This statement awards loyalty points to the customer who placed a specific order. The calculation adds five percent of the order total, rounded down to the nearest whole number.  
**See Appendix B (DML.sql) for the complete script.**

Custom DML 2: Receiving Inventory for a Product

When a shipment arrives, on-hand quantity must be increased. This statement adds forty units to the specified product.  
**See Appendix B (DML.sql) for the complete script.**

Custom DML 3: Correcting Supplier Contact Information

Supplier contact details change over time. This statement updates the contact email for a single supplier.  
**See Appendix B (DML.sql) for the complete script.**

Reading and Reporting With SELECT Statements

This reporting section explains, in plain language, how the queries read data and turn it into clear information for daily decisions. Each report uses simple relational concepts such as grouping, joins, and aggregate functions to roll line items into totals that staff and managers can understand at a glance. Revenue is calculated from order details as quantity multiplied by unit price, and grouping is used to summarize by month, customer, product, or category as needed (Coronel & Morris, 2018; Harrington, 2016; IBM, 2023).  
**See Appendix C (REPORT.sql) for all reporting queries, including the required crosstab.**

The first report produces revenue per month grouped by customers. It joins orders to order details and customers, calculates revenue as quantity times unit price, formats the order date by year and month, and then groups the results by that month and by the customer. The outcome is a monthly view of what each customer spent. Store staff can use this to recognize loyal customers, plan thank you messages and decide who might benefit from loyalty rewards or targeted invitations to workshops. Managers can also see seasonality in customer behavior and plan staffing accordingly (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix C (REPORT.sql) for the complete query.**

The second report produces revenue per month grouped by product. It joins orders to order details and products, calculates revenue the same way, and groups by year and month together with the product. The result shows which items drive sales in each month. Buyers and department leads can use this to guide reordering, pricing reviews, and end cap displays. If a product spikes in a given month, the team can check the associated marketing activity or workshop topics to see what helped performance and repeat the approach where appropriate (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix C (REPORT.sql) for the complete query.**

The third report counts products by category. It joins categories to products and uses a count to show how many active items appear in each category. This helps maintain a balanced catalog and reveals gaps. For example, if a gluten free bakery has many items while dairy alternatives is thin, the team can add options where customers need more choice. The same result also helps with shelf planning and planogram updates because managers can see the size of each category in a single view (Coronel & Morris, 2018; IBM, 2023).  
**See Appendix C (REPORT.sql) for the complete query.**

The first custom report is a low stock reorder list. It reads the products table, compares the current stock level with the reorder level, and returns items that are at or below that threshold. The query also includes the supplier and category so that a buyer can act without opening other screens. This report supports inventory accuracy and prevents missed sales by prompting timely replenishment. It also improves vendor communication because the supplier’s name and contact information appear directly in the result set (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix C (REPORT.sql) for the complete query.**

The second custom report lists the top customers by revenue for the year. It aggregates line items by customer across the selected year and sorts by the total in descending order. The result highlights the customers who contribute the most to yearly revenue. The store can use this information to plan appreciation messages, exclusive previews, or loyalty offers that match each customer’s preferences. This report also supports staff coaching by showing which customer relationships are strongest and where service follow up could deepen engagement (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix C (REPORT.sql) for the complete query.**

The third custom report is a crosstab of monthly revenue by category. The query keeps the data in a single pass by using a set of conditional sums. For each month, the report sums revenue for that month and places the total in its own column. The result is a single table where each row is a category and each column is a month. Managers can scan the row for a category and see its month over month pattern without scrolling through separate reports. This format is useful for leadership meetings because it reads like a dashboard but remains grounded in straightforward SQL logic that is easy to audit and maintain (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix C (REPORT.sql) for the complete query.**

Together, these reports turn raw transactions into practical insights. Frontline staff can quickly answer questions about what sold, who bought it, and what needs to be reordered. These reports will enable the managers to track category strength, identify the most engaged customers, and spot seasonal patterns that affect staffing and purchasing. Since the queries rely on core relational features, they remain easy to test and adapt as the store’s needs evolve (Coronel & Morris, 2018; Harrington, 2016; IBM, 2023).

How SQL Implementation Supports the Mission and Goals

The database structure created in this phase supports the store mission by turning day-to-day activity into reliable information that staff can trust. The definition scripts establish clear tables for customers, products, orders, and workshops so that every record is stored once and linked correctly. This design reduces errors, speeds up lookups at the counter, and prevents confusion about where information belongs. Normalized tables and enforced keys protect data quality and create a single source of truth that managers and team members can use without second guessing the results (Coronel & Morris, 2018; Harrington, 2016).

The data manipulation scripts reflect real store tasks and therefore support operations directly. The delete by order script removes an order safely by clearing its line items first, which keeps relationships intact and avoids broken references. The price update script supports routine vendor and market changes without side effects because it targets one product by its unique identifier. The custom scripts mirror common workflows. The loyalty points script rewards customers based on what they purchased. The inventory receipt script increases on hand stock when shipments arrive. The supplier contacts update keeps purchasing information current so buyers can act quickly. Each script helps staff complete important work with a single, precise statement rather than manual edits that can introduce mistakes (Coronel & Morris, 2018; IBM, 2023).

The reporting queries translate raw transactions into information that guides decisions. Monthly revenue by customers shows engagement patterns and helps the team tailor outreach and education. Monthly revenue by product reveals which items carry the month so that buyers can plan promotions and reorder schedules with confidence. The count of products by category helps maintain a balanced assortment that matches customer demand. The low stock list prevents missed sales by prompting timely replenishment and includes supplier contact details to shorten the time from insight to action.

The top customers report supports recognition and retention efforts. The category crosstab places monthly totals in a compact, friendly meeting format so leaders can scan trends briefly. Together, these reports connect daily work to measurable goals such as revenue growth, product availability, and customer education outcomes (Coronel & Morris, 2018; Harrington, 2016; IBM, 2023).

This implementation also aligns with the store’s goal to serve the community through trustworthy wellness education. The workshops’ tables capture planned classes, the staff who lead them, and customer attendance. The same relational tools that power sales reporting also make it possible to see which topics draw interest, which time slots fill first, and which customers return for more learning. That insight supports better scheduling, stronger content planning, and timely follow up after events. In practical terms, the database turns workshops from a manual signup sheet into a measurable part of the mission that grows alongside the retail operation.

Finally, the scripts are written in a clear and repeatable format so nontechnical staff can run them with simple verification queries. Each statement is narrow in scope, uses plain identifiers, and returns results that are easy to confirm. That approach reduces training time and helps the store avoid dependence on a single technical specialist. The database therefore becomes a daily tool for the entire team, not just a back-office system, which is consistent with best practice guidance for relational design that emphasizes clarity, consistency, and direct support for business decisions (Coronel & Morris, 2018; Harrington, 2016).  
**See Appendix A (DDL.sql) for the complete database build script and sample data.**  
**See Appendix B (DML.sql) for all data manipulation scripts.**  
**See Appendix C (REPORT.sql) for all reporting and crosstab queries.**

References

Coronel, C., & Morris, S. (2018). *Database systems: Design, implementation, and management* (13th ed.). Cengage Learning.

Forbes Technology Council. (2021, June 3). Why a good database management system is essential for small business. *Forbes.* <https://www.forbes.com/sites/forbestechcouncil/2021/06/03/why-a-good-database-management-system-is-essential-for-small-business>

Harrington, J. L. (2016). *Relational database design and implementation* (4th ed.). Morgan Kaufmann.

IBM. (2023). What is a relational database? <https://www.ibm.com/topics/relational-databases>

Lucidchart. (2024). Entity relationship diagram (ERD) guide. <https://www.lucidchart.com/pages/er-diagrams>

Rigby, D. K., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard Business Review.* <https://hbr.org/2018/05/agile-at-scale>

Salesforce. (2023). What is CRM? *Salesforce.* <https://www.salesforce.com/learning-centre/crm/what-is-crm/>

Shopify. (2022). The future of retail: 2022 global report. <https://www.shopify.com/research/future-of-commerce>

TRACTIAN. (n.d.). Inventory management software. <https://tractian.com/en/cmms/inventory-management-software?utm_campaign=usa-consideration-tracos-cold-search-bing&utm_term=kw-inventory-management&utm_content=inventory%20management%20system&msclkid=450c252d6a9a17e6f4834da839c02354&utm_source=bing&utm_medium=cpc>

# Appendix A: DDL.sql

This script creates the database objects for Marc’s Health Food Store and loads twenty sample rows per table. Run the entire block once in your MySQL editor.

-- DDL.sql: Create and populate database schema for Marc's Health Food Store

-- Drop tables in reverse order to avoid foreign key conflicts

IF OBJECT\_ID('workshop\_attendance', 'U') IS NOT NULL DROP TABLE workshop\_attendance;

IF OBJECT\_ID('wellness\_workshops', 'U') IS NOT NULL DROP TABLE wellness\_workshops;

IF OBJECT\_ID('order\_details', 'U') IS NOT NULL DROP TABLE order\_details;

IF OBJECT\_ID('orders', 'U') IS NOT NULL DROP TABLE orders;

IF OBJECT\_ID('products', 'U') IS NOT NULL DROP TABLE products;

IF OBJECT\_ID('employees', 'U') IS NOT NULL DROP TABLE employees;

IF OBJECT\_ID('customers', 'U') IS NOT NULL DROP TABLE customers;

IF OBJECT\_ID('suppliers', 'U') IS NOT NULL DROP TABLE suppliers;

IF OBJECT\_ID('categories', 'U') IS NOT NULL DROP TABLE categories;

-- Create parent tables

CREATE TABLE categories (

    category\_id INT PRIMARY KEY IDENTITY(1,1),

    category\_name VARCHAR(100)

);

CREATE TABLE suppliers (

    supplier\_id INT PRIMARY KEY IDENTITY(1,1),

    supplier\_name VARCHAR(100),

    contact\_info VARCHAR(100)

);

CREATE TABLE customers (

    customer\_id INT PRIMARY KEY IDENTITY(1,1),

    full\_name VARCHAR(100),

    email VARCHAR(100),

    phone VARCHAR(20),

    dietary\_preference VARCHAR(100),

    loyalty\_points INT

);

CREATE TABLE employees (

    employee\_id INT PRIMARY KEY IDENTITY(1,1),

    full\_name VARCHAR(100),

    role VARCHAR(60)

);

-- Create child tables

CREATE TABLE products (

    product\_id INT PRIMARY KEY IDENTITY(1,1),

    product\_name VARCHAR(120),

    price DECIMAL(8,2),

    stock\_level INT,

    reorder\_level INT,

    category\_id INT,

    supplier\_id INT,

    FOREIGN KEY (category\_id) REFERENCES categories(category\_id),

    FOREIGN KEY (supplier\_id) REFERENCES suppliers(supplier\_id)

);

CREATE TABLE orders (

    order\_id INT PRIMARY KEY IDENTITY(101,1),

    order\_date DATE,

    customer\_id INT,

    employee\_id INT,

    FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id),

    FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)

);

CREATE TABLE order\_details (

    order\_id INT,

    product\_id INT,

    quantity INT,

    unit\_price DECIMAL(8,2),

    PRIMARY KEY (order\_id, product\_id),

    FOREIGN KEY (order\_id) REFERENCES orders(order\_id),

    FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);

CREATE TABLE wellness\_workshops (

    workshop\_id INT PRIMARY KEY IDENTITY(1,1),

    title VARCHAR(120),

    scheduled\_date DATE,

    capacity INT,

    employee\_id INT,

    FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)

);

CREATE TABLE workshop\_attendance (

    attendance\_id INT PRIMARY KEY IDENTITY(1,1),

    customer\_id INT,

    workshop\_id INT,

    FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id),

    FOREIGN KEY (workshop\_id) REFERENCES wellness\_workshops(workshop\_id)

);

-- Insert 20 records into each table

INSERT INTO categories (category\_name) VALUES

('Organic Produce'), ('Bulk Grains and Legumes'), ('Plant-Based Proteins'), ('Herbal Supplements'),

('Cold Beverages and Kombucha'), ('Hot Beverages and Tea'), ('Gluten-Free Bakery'), ('Nut Butters and Spreads'),

('Dairy Alternatives'), ('Healthy Snacks and Bars'), ('Cooking Oils and Vinegars'), ('Spices and Seasonings'),

('Natural Sweeteners'), ('Personal Care and Beauty'), ('Household and Cleaning'), ('Frozen Fruits and Veggies'),

('Fermented Foods'), ('Breakfast and Cereals'), ('Hydration and Electrolytes'), ('Kids Health and Lunchbox');

INSERT INTO suppliers (supplier\_name, contact\_info) VALUES

('Crimson Creek Provisions', 'orders@crimsoncreek.com'), ('Twilight Roots', 'support@twilightroots.com'),

('Velvet Valley Naturals', 'contact@velvetvalleynaturals.com'), ('Blue Fern Co.', 'hello@bluefernco.com'),

('Cedar & Sage', 'sales@cedarsage.com'), ('Echo Harvest Supply', 'info@echoharvest.com'),

('Wild Ember Foods', 'admin@wildember.com'), ('Moss & Moon Organics', 'mossmoon@suppliers.com'),

('WanderRoot Partners', 'orders@wanderroot.com'), ('Clover Canyon Distributors', 'service@clovercanyon.com'),

('FloraRise Co.', 'contact@florarise.com'), ('Dune Bloom Supply', 'orders@dunebloom.com'),

('HearthGlow Market', 'sales@hearthglow.com'), ('Brassleaf Naturals', 'brassleaf@suppliers.com'),

('Kindled Grove Co.', 'hello@kindledgrove.com'), ('Crescent Acres', 'support@crescentacres.com'),

('Nimbus Nest Supply', 'orders@nimbusnest.com'), ('Ambertrail Foods', 'info@ambertrail.com'),

('Hollow Pine Provisions', 'contact@hollowpine.com'), ('Sundew Farms Collective', 'admin@sundewfarms.com');

INSERT INTO customers (full\_name, email, phone, dietary\_preference, loyalty\_points) VALUES

('Dana Jones', 'dana@example.com', '555-0110', 'Vegan', 20), ('Nailah Smith', 'nailah@example.com', '555-0111', 'Gluten-Free', 35),

('Orion Sky', 'orion.sky@example.com', '555-0112', 'Paleo', 15), ('River Lynn', 'river.lynn@example.com', '555-0113', 'Vegetarian', 12),

('Luna Vance', 'luna.vance@example.com', '555-0114', 'Vegan', 30), ('Zane Cruz', 'zane.cruz@example.com', '555-0115', 'Keto', 8),

('Nova Reese', 'nova.reese@example.com', '555-0116', 'Dairy-Free', 25), ('Elias Reed', 'elias.reed@example.com', '555-0117', 'None', 5),

('Sage Monroe', 'sage.monroe@example.com', '555-0118', 'Gluten-Free', 40), ('Indigo Vale', 'indigo.vale@example.com', '555-0119', 'Vegan', 18),

('Kai Bennett', 'kai.bennett@example.com', '555-0120', 'Low-Sodium', 10), ('Skye Emerson', 'skye.emerson@example.com', '555-0121', 'Vegetarian', 22),

('Phoenix Lane', 'phoenix.lane@example.com', '555-0122', 'Keto', 16), ('Jasper Storm', 'jasper.storm@example.com', '555-0123', 'None', 12),

('Azaria Blue', 'azaria.blue@example.com', '555-0124', 'Vegan', 33), ('Milo Dash', 'milo.dash@example.com', '555-0125', 'Dairy-Free', 11),

('Zuri West', 'zuri.west@example.com', '555-0126', 'Gluten-Free', 27), ('Lyric Moon', 'lyric.moon@example.com', '555-0127', 'Low-Carb', 14),

('Calla Brooks', 'calla.brooks@example.com', '555-0128', 'Vegetarian', 19), ('Sol Eden', 'sol.eden@example.com', '555-0129', 'None', 9);

INSERT INTO employees (full\_name, role) VALUES

('Sage Moon', 'Cashier'), ('River Knight', 'Stock Associate'), ('Lyric Hart', 'Nutrition Specialist'),

('Zion Wilder', 'Floor Manager'), ('Nova Quinn', 'Cashier'), ('Cypress Lane', 'Shift Supervisor'),

('Indigo Ray', 'Marketing Assistant'), ('Phoenix Snow', 'Stock Associate'), ('Skye Ocean', 'Nutrition Specialist'),

('Rowan Blaze', 'Inventory Coordinator'), ('Juniper Blue', 'Cashier'), ('Orion Steele', 'Floor Manager'),

('Vesper Rain', 'Workshop Coordinator'), ('Zephyr Cloud', 'Nutrition Specialist'), ('Solstice Day', 'Customer Service'),

('Atlas Vale', 'Maintenance'), ('Echo Star', 'Cashier'), ('Draven Ash', 'Shift Supervisor'),

('Tansy Wren', 'Workshop Assistant'), ('Lazlo Frost', 'Logistics Support');

INSERT INTO products (product\_name, price, stock\_level, reorder\_level, category\_id, supplier\_id) VALUES

('Sunrise Kale Bunch', 2.49, 85, 20, 1, 1), ('Canyon Chickpea Dry Pack 2 lb', 4.79, 120, 30, 2, 10),

('Maple Herb Tempeh 8 oz', 5.99, 60, 15, 3, 7), ('Calm Focus Ashwagandha 90 ct', 16.95, 40, 10, 4, 14),

('Citrus Ginger Kombucha 16 oz', 3.49, 140, 35, 5, 2), ('Cinnamon Chai Wellness Tea 20 ct', 6.25, 75, 18, 6, 11),

('Almond Flour Morning Muffin 4 pack', 7.49, 32, 8, 7, 13), ('Stoneground Almond Butter 12 oz', 8.99, 50, 12, 8, 15),

('Barista Oat Milk 32 oz', 3.29, 110, 28, 9, 16), ('Trail Glow Nut Bar Single', 1.99, 200, 50, 10, 18),

('Cold-Pressed Avocado Oil 16 oz', 12.95, 45, 12, 11, 19), ('Smoked Paprika Glass Jar 2 oz', 4.25, 90, 22, 12, 4),

('Wildflower Honey 12 oz', 7.75, 55, 14, 13, 20), ('Lavender Shea Hand Cream 3 oz', 9.50, 38, 10, 14, 5),

('Citrus Thyme Cleaner Refill 32 oz', 6.99, 62, 16, 15, 6), ('Frozen Mango Chunks 16 oz', 3.89, 130, 32, 16, 8),

('Classic Kimchi Pint', 5.49, 44, 12, 17, 9), ('Ancient Grain Granola 12 oz', 6.49, 72, 18, 18, 12),

('Cucumber Mint Electrolyte Drink', 2.29, 150, 36, 19, 17), ('Kids Apple Cinnamon Bites 5 ct', 4.59, 58, 14, 20, 3);

INSERT INTO orders (order\_date, customer\_id, employee\_id) VALUES

('2025-01-03', 1, 2), ('2025-01-04', 3, 4), ('2025-01-05', 5, 1), ('2025-01-06', 2, 3),

('2025-01-08', 4, 5), ('2025-01-09', 6, 2), ('2025-01-10', 8, 1), ('2025-01-12', 10, 3),

('2025-01-14', 7, 4), ('2025-01-15', 9, 5), ('2025-02-01', 1, 2), ('2025-02-02', 3, 4),

('2025-02-03', 5, 1), ('2025-02-05', 2, 3), ('2025-02-07', 4, 5), ('2025-02-08', 6, 2),

('2025-02-09', 8, 1), ('2025-02-10', 10, 3), ('2025-02-12', 7, 4), ('2025-02-14', 9, 5);

INSERT INTO order\_details (order\_id, product\_id, quantity, unit\_price) VALUES

(101, 5, 2, 3.49), (102, 1, 3, 2.49), (103, 8, 1, 8.99), (104, 12, 2, 4.25),

(105, 3, 2, 5.99), (106, 11, 1, 12.95), (107, 9, 3, 3.29), (108, 14, 1, 9.50),

(109, 7, 2, 7.49), (110, 16, 4, 3.89), (111, 19, 2, 2.29), (112, 6, 1, 6.25),

(113, 13, 1, 7.75), (114, 17, 2, 5.49), (115, 2, 2, 4.79), (116, 15, 1, 6.99),

(117, 18, 1, 6.49), (118, 4, 1, 16.95), (119, 20, 2, 4.59), (120, 10, 5, 1.99);

INSERT INTO wellness\_workshops (title, scheduled\_date, capacity, employee\_id) VALUES

('Gut Health 101: Fermented Foods at Home', '2025-03-05', 20, 13),

('Plant Powered Meal Prep for Busy Weeks', '2025-03-08', 24, 3),

('Intro to Herbal Immunity Support', '2025-03-12', 18, 14),

('Low Sugar Baking Basics', '2025-03-15', 16, 7),

('Smoothie Lab: Protein and Greens', '2025-03-19', 22, 9),

('Reading Labels: Spotting Hidden Ingredients', '2025-03-22', 25, 12),

('Family Lunchbox Makeover', '2025-03-26', 20, 15),

('Tea Rituals for Calm and Focus', '2025-03-29', 18, 11),

('Gluten Free Kitchen Setup', '2025-04-02', 20, 14),

('Keto Friendly Pantry Staples', '2025-04-05', 18, 3),

('Budget Friendly Organic Shopping', '2025-04-09', 26, 12),

('Anti Inflammatory Cooking Made Simple', '2025-04-12', 22, 9),

('Electrolytes and Hydration for Spring Training', '2025-04-16', 28, 17),

('Fermentation: Kimchi and Kraut Workshop', '2025-04-19', 16, 13),

('Dairy Free Desserts that Satisfy', '2025-04-23', 20, 5),

('Spice Cabinet Masterclass', '2025-04-26', 18, 12),

('Breakfast Reset: High Fiber Starts', '2025-04-30', 24, 3),

('Kids in the Kitchen: Safe Knife Skills', '2025-05-03', 14, 20),

('Spring Clean Your Home the Non Toxic Way', '2025-05-07', 22, 6),

('Vegan Protein Three Ways: Tempeh, Tofu, and Beans', '2025-05-10', 20, 3);

INSERT INTO workshop\_attendance (customer\_id, workshop\_id) VALUES

(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8), (9, 9), (10, 10),

(1, 11), (2, 12), (3, 13), (4, 14), (5, 15), (6, 16), (7, 17), (8, 18), (9, 19), (10, 20);

-- Sanity checks (optional, for verification)

SELECT COUNT(\*) AS categories\_rows FROM categories;

SELECT COUNT(\*) AS suppliers\_rows FROM suppliers;

SELECT COUNT(\*) AS customers\_rows FROM customers;

SELECT COUNT(\*) AS employees\_rows FROM employees;

SELECT COUNT(\*) AS products\_rows FROM products;

SELECT COUNT(\*) AS orders\_rows FROM orders;

SELECT COUNT(\*) AS order\_details\_rows FROM order\_details;

SELECT COUNT(\*) AS workshops\_rows FROM wellness\_workshops;

SELECT COUNT(\*) AS attendance\_rows FROM workshop\_attendance;

# Appendix B: DML.sql

This script includes the required delete and update tasks and three custom tasks. Each block can run on its own.

-- DML.sql: Data manipulation scripts for Marc's Health Food Store

-- Delete an entire order by order\_id

BEGIN TRANSACTION;

DELETE FROM order\_details WHERE order\_id = 111;

DELETE FROM orders WHERE order\_id = 111;

COMMIT;

-- Verify

SELECT \* FROM orders WHERE order\_id = 111;

SELECT \* FROM order\_details WHERE order\_id = 111;

-- Update product price by product\_id

UPDATE products

SET price = 6.79

WHERE product\_id = 18;

-- Verify

SELECT product\_id, product\_name, price

FROM products

WHERE product\_id = 18;

-- Custom DML 1: Award loyalty points from order total (5% of total, rounded down)

UPDATE customers c

SET c.loyalty\_points = c.loyalty\_points + FLOOR(t.order\_total \* 0.05)

FROM (

SELECT o.customer\_id, SUM(od.quantity \* od.unit\_price) AS order\_total

FROM orders o

JOIN order\_details od ON od.order\_id = o.order\_id

WHERE o.order\_id = 113

GROUP BY o.customer\_id

) t

WHERE c.customer\_id = t.customer\_id;

-- Verify

SELECT c.customer\_id, c.full\_name, c.loyalty\_points

FROM customers c

WHERE c.customer\_id IN (SELECT customer\_id FROM orders WHERE order\_id = 113);

-- Custom DML 2: Receive inventory for a product (add 40 units)

UPDATE products

SET stock\_level = stock\_level + 40

WHERE product\_id = 9;

-- Verify

SELECT product\_id, product\_name, stock\_level

FROM products

WHERE product\_id = 9;

-- Custom DML 3: Correct supplier contact information

UPDATE suppliers

SET contact\_info = 'service@cedarsage.com'

WHERE supplier\_id = 5;

-- Verify

SELECT supplier\_id, supplier\_name, contact\_info

FROM suppliers

WHERE supplier\_id = 5;

# Appendix C: REPORT.sql

This script includes the three required reports and three custom reports, including a crosstab.

-- REPORT.sql: Reporting queries for Marc's Health Food Store

-- Report 1: Revenue per month grouped by customer

SELECT FORMAT(o.order\_date, 'yyyy-MM') AS order\_month,

c.customer\_id, c.full\_name,

ROUND(SUM(od.quantity \* od.unit\_price), 2) AS revenue

FROM orders o

JOIN order\_details od ON od.order\_id = o.order\_id

JOIN customers c ON c.customer\_id = o.customer\_id

WHERE YEAR(o.order\_date) = 2025

GROUP BY FORMAT(o.order\_date, 'yyyy-MM'), c.customer\_id, c.full\_name

ORDER BY order\_month, revenue DESC;

-- Report 2: Revenue per month grouped by product

SELECT FORMAT(o.order\_date, 'yyyy-MM') AS order\_month,

p.product\_id, p.product\_name,

ROUND(SUM(od.quantity \* od.unit\_price), 2) AS revenue

FROM orders o

JOIN order\_details od ON od.order\_id = o.order\_id

JOIN products p ON p.product\_id = od.product\_id

WHERE YEAR(o.order\_date) = 2025

GROUP BY FORMAT(o.order\_date, 'yyyy-MM'), p.product\_id, p.product\_name

ORDER BY order\_month, revenue DESC;

-- Report 3: Product count grouped by category

SELECT c.category\_id, c.category\_name, COUNT(p.product\_id) AS product\_count

FROM categories c

LEFT JOIN products p ON p.category\_id = c.category\_id

GROUP BY c.category\_id, c.category\_name

ORDER BY product\_count DESC, c.category\_name;

-- Custom Report 1: Low stock reorder list

SELECT p.product\_id, p.product\_name, p.stock\_level, p.reorder\_level,

cat.category\_name, s.supplier\_name, s.contact\_info

FROM products p

JOIN categories cat ON cat.category\_id = p.category\_id

JOIN suppliers s ON s.supplier\_id = p.supplier\_id

WHERE p.stock\_level <= p.reorder\_level

ORDER BY p.stock\_level ASC, p.product\_name;

-- Custom Report 2: Top customers by revenue in 2025

SELECT TOP 5 c.customer\_id, c.full\_name,

ROUND(SUM(od.quantity \* od.unit\_price), 2) AS total\_revenue

FROM customers c

JOIN orders o ON o.customer\_id = c.customer\_id

JOIN order\_details od ON od.order\_id = o.order\_id

WHERE YEAR(o.order\_date) = 2025

GROUP BY c.customer\_id, c.full\_name

ORDER BY total\_revenue DESC;

-- Custom Report 3: CROSSTAB monthly revenue by category

SELECT cat.category\_name,

ROUND(SUM(CASE WHEN YEAR(o.order\_date) = 2025 AND MONTH(o.order\_date) = 1

THEN od.quantity \* od.unit\_price ELSE 0 END), 2) AS Jan\_2025,

ROUND(SUM(CASE WHEN YEAR(o.order\_date) = 2025 AND MONTH(o.order\_date) = 2

THEN od.quantity \* od.unit\_price ELSE 0 END), 2) AS Feb\_2025,

ROUND(SUM(CASE WHEN YEAR(o.order\_date) = 2025 AND MONTH(o.order\_date) = 3

THEN od.quantity \* od.unit\_price ELSE 0 END), 2) AS Mar\_2025,

ROUND(SUM(CASE WHEN YEAR(o.order\_date) = 2025 AND MONTH(o.order\_date) = 4

THEN od.quantity \* od.unit\_price ELSE 0 END), 2) AS Apr\_2025

FROM categories cat

JOIN products p ON p.category\_id = cat.category\_id

JOIN order\_details od ON od.product\_id = p.product\_id

JOIN orders o ON o.order\_id = od.order\_id

GROUP BY cat.category\_name

ORDER BY cat.category\_name;