PROGRAMMING FOR BUSINESS ANALYTICS

KARAN RAJESH LOKARE

240753205

Table of Contents

Introduction: Business Scenario and Application

Our application helps small online retail companies sell consumer goods such as footwear, apparel, and accessories(E-commerce Support Services). The growing business needs solutions for handling orders, better monitoring products, and keeping customers informed. The solution connects a relational database to VBA applications and Excel UI to build a system that grows efficiently.

Our system aims to improve basic business operations by managing orders better while tracking inventory and reporting sales data. The system links Access database production with Excel data entry and reporting to provide instant updates and automated reporting capabilities. Our application functions better through VBA subroutines, which let us import data of customers

Our system addresses main business problems by fixing human mistakes, simplifying data management, and helping managers make better choices. The system shows all waiting orders quickly while watching inventory and lets users see sales results in special data views. Better daily operations become more efficient while our long-term business success increases as we make decisions based on data. The system helps small businesses build strong financial management practices at budget-friendly prices.

Database: Description and Queries

This project uses a database design that helps run business activities smoothly. It consists of three interconnected tables. Our system depends on three basic components: Customers, Products, and Orders. The database uses relationships to keep data safe and allows us to merge related information precisely. In the database system, the CustomerID column from the Customers table functions as a foreign key in the Orders table, while the ProductID column from the Products table does the same in the Orders table. A relational database design helps maintain correct data while making searches run faster.

The Customers table stores basic customer information following these fields: CustomerID (integer), Name (text), Email (text), PhoneNumber (text), and RegistrationDate (date). Its basic role is to keep customer data so staff can match orders back to individual shoppers. The database stores product data through ProductID (number), Name (string), Price (decimal), Category (string), and Stock (number). The table assists companies in controlling their stock supplies while showing product prices (Rodrigues *et al*., 2022). The Orders table contains OrderID and CustomerID numbers along with ProductID numbers, Quantity ordered,

OrderDate and Status text details. This tool both documents business deals and shows how orders move toward completion.

In this project, I created SQL queries to analyse product and sales data. One query organizes products by price, making it easier to review pricing trends and identify items in different price ranges. Another query groups products by category, providing a clear view of how inventory is distributed. Lastly, I calculated revenue for each product category, giving insights into which categories generate the most income. These queries work together to provide a detailed understanding of product performance and sales.

Front-end: Design and Business Use of the Excel Sheet

The front-end design of my project comprises FOUR main sheets, each serving a distinct purpose:

1. **Import Sheet**: This sheet is dedicated to importing data from the database and generating a detailed customer report. Based on suggestions, the VBA code automatically creates a new sheet where key customer metrics are summarized.
2. **Load & Search Sheet**: This sheet allows users to load data from the database and search for specific customer names. If a match is found, the corresponding row is highlighted, providing a simple and efficient way to locate records.
3. **Conditional Formatting Sheet**: This sheet utilizes conditional formatting to enhance data readability. For example, cells with the value "Delivered" are highlighted in blue, making it easy to identify the status of orders.
4. **Pivot Table Analysis Sheet:** In addition to these sheets, the project includes a Pivot Table Analysis sheet, which provides valuable insights through dynamic data summaries:

- **Product Pricing Summary:** This pivot table displays the total price of various products and includes a filter for Product ID, enabling users to analyze specific items effectively.
- **Product Performance Overview:** Another pivot table summarizes critical metrics, such as the total quantity, order IDs, and prices for each product name. This allows users to evaluate product performance comprehensively

.

VBA Middleware: Description of the VBA Code

The VBA middleware acts as the central piece of our system to connect the database and Excel user interface. The system uses VBA to automate data tasks and simplify user work so users can efficiently run essential business processes (Chu and Yong., 2021). The middleware uses VBA to run built-in programs that make database data simpler to access and maintain.

The VBA code comprises FOUR primary subroutines:

1. GenerateReport - This section is used to create a summary report with important customer metrics. When you run this, it generates a new worksheet called "Report" (or uses an existing one if it's already there) and adds details like:

- The **total number of customers** in your dataset, helping to understand the scale of your customer base.
- The **earliest and latest registration dates**, which can give you insights into customer sign-up trends over time.

This report helps you quickly get an overview of your data without needing to dive into each individual record.

2. ImportDataFromAccess - This part is all about bringing in data from an Access database. It connects to a file (like RETAIL.accdb) and lets you input a custom SQL query. After the query is run, the data is pulled into the "IMPORT" sheet. This gives you the flexibility to import fresh data directly into Excel without manually entering it, ensuring you're always working with the most up-to-date information.

3. SearchCustomerByName - This feature makes it easy to search for a specific customer by name in the "LOAD&SEARCH" sheet. When you enter the name, it looks for an exact match, and if found, it highlights the entire row where that customer's information is located. This is super helpful when you need to quickly find and focus on a particular customer without scrolling through the entire list.

4. LoadFromDatabase - The LoadFromDatabase part is designed to pull customer data from the database into Excel. It connects to your Access database (RETAIL.accdb) and fetches all the customer records by running a query like SELECT * FROM Customers. Once the data is

retrieved, it automatically loads it into the "LOAD&SEARCH" worksheet. If there’s any previous data on the sheet, it clears that first to avoid clutter, so you always get the latest customer information. This makes it easy to refresh your data whenever needed.

Conclusion: Scaling the Application for a Real Business

A database system can deal with real business needs when we add extra tables to handle supplier, product, and customer information. Our system can save operation time by automatically sending customers order status updates and notifications. To manage larger amounts of data, the database system needs an advanced version that guarantees better growth and better processing speed. By connecting to CRM tools and e-commerce platforms, a full enterprise system would let data move easily between applications and operations run more smoothly.