# Inventory and Supplier Management System

***SQL Project Report – Elevate Lab Internship***

**Devu S**
College of Engineering Chengannur

---

## Abstract

This project was developed as part of the Elevate Lab SQL Internship program. The aim was to design and implement a structured and functional **Inventory and Supplier Management System** using SQL. The system supports tracking of products, managing suppliers, handling stock updates, and recording purchases. The project demonstrates key database concepts such as normalization, foreign keys, joins, stored procedures, views, and functions.

---

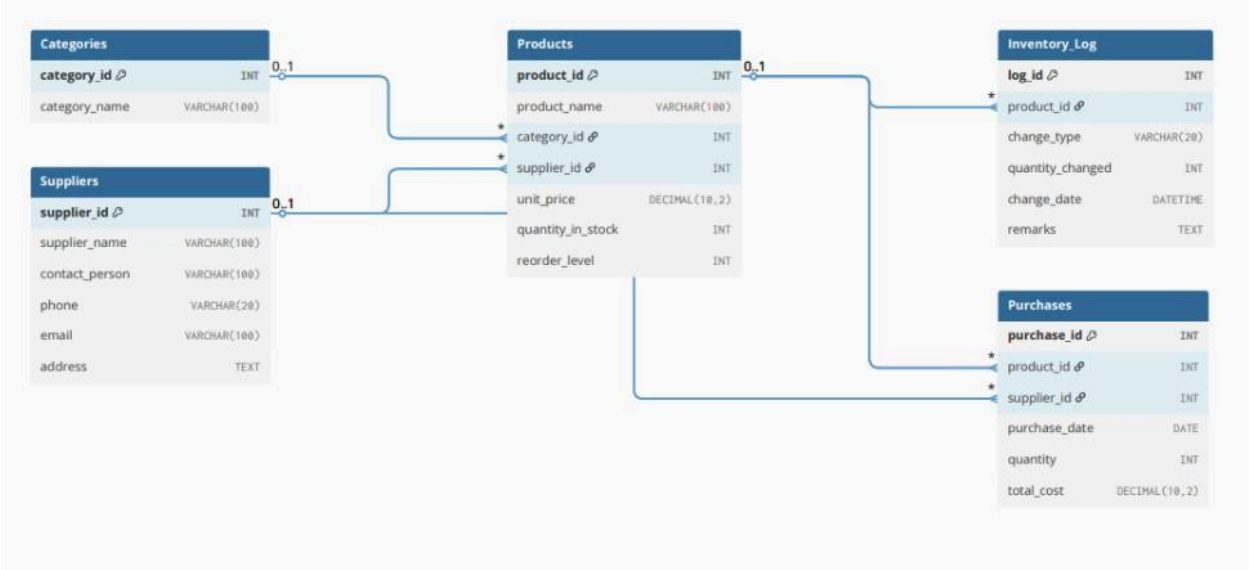
## Introduction

Efficient inventory and supplier management plays a pivotal role in ensuring the smooth operation of businesses that rely on physical goods. Whether it is a retail store, a manufacturing unit, or a supply chain system, maintaining accurate records of stock levels and supplier details is crucial for cost control, timely replenishment, and customer satisfaction.

This project aims to develop a robust backend system for managing inventory and supplier operations using **Structured Query Language (SQL)**. The system is designed to store, organize, and retrieve data related to product categories, supplier profiles, stock quantities, purchase transactions, and stock movement logs. It leverages relational database design principles to maintain data consistency and minimize redundancy through normalization.

By implementing a carefully planned schema with properly linked tables, the system allows seamless insertion, updating, and retrieval of data. It also supports analytical queries and reports through views, functions, and stored procedures, enabling stakeholders to monitor low stock levels, track purchase history, and generate summaries.

# ER Diagram




---

# ER Diagram Summary

The database schema consists of five core entities:

- **Categories** – Defines types of products (e.g., Electronics, Stationery)
- **Suppliers** – Stores supplier information such as contact and address
- **Products** – Tracks each item with stock level, unit price, and reorder threshold
- **Purchases** – Records of purchase transactions from suppliers
- **Inventory_Log** – History of stock additions/removals with reasons and timestamps

Each product is linked to a category and a supplier. Purchases are logged with timestamps, and inventory changes are recorded separately for traceability.

---

# Table Descriptions

- **Categories**: Stores types of items like Electronics, Stationery, etc.

- **Suppliers**: Stores supplier names, contact details, and locations.
- **Products**: Each row includes product name, category, supplier, unit price, quantity in stock, and reorder level.
- **Purchases**: Logs each transaction with product ID, quantity, supplier, total cost, and date.
- **Inventory_Log**: Tracks additions or reductions in stock with change type, quantity, date, and remarks.

---

## SQL Highlights

- ✅ **Data Insertion** using full rows, partial data with `DEFAULT`, and handling of `NULL` values.
- 🔍 **Queries** using JOINs, filtering, ordering, aggregation, and subqueries.
- 📊 **Views** for simplified reporting like `LowStockProducts` and monthly summaries.
- 📈 **Functions** to compute total stock value and check restocking needs.
- 🔁 **Procedures** to automate purchase logging and update stock levels.
- 🔐 **Integrity** ensured with foreign key relationships and consistent schema design.

---

## Conclusion

This project allowed me to implement a real-world use case of database design using SQL. I gained practical experience in writing queries, managing data, and using SQL features such as stored procedures, views, and functions. I also learned how a normalized relational schema can be used to support decision-making in inventory and supplier workflows. This project strengthened my understanding of backend data systems and how they support business logic efficiently.