



## **Advanced Database Management System ( For Jalitha Harware )**

**ADBMS Final Assignment 2025  
Module Code : 304.3**

**Group T**

<b>Student ID</b>	<b>Name</b>
28666	W L C Wickramasinghe
28466	R P J D Rajapaksha
28510	K L D P P Kakulandala
28663	M A J C B Dissanayake
28131	H A J N Heshani
28230	W H M N Jayawardhana
28548	K S T Kavindi
27595	H K I Dhananjaya
28372	R V I Rathnayake
28340	U V N S Mayuranga

# Organization Details

- **Business Name:** Jalitha Hardware  
**Owner:** R.P.K. Rajapaksha
- **Overview:**

Jalitha Hardware Shop is a local retail store focused on construction materials and hardware supplies. The shop offers a wide range of products such as cement, bricks, tools, plumbing items, and electrical goods to meet the needs of construction professionals and DIY customers.
- **Operations:**

Owned and operated by R.P.K. Rajapaksha, the shop provides quality products and personalized customer service. It caters mainly to local builders, contractors, and homeowners, ensuring timely availability of essential materials.
- **Goal:**
  1. **Customer Trust:**

establish the shop as the most trusted source for construction and hardware supplies in the local area.
  2. **Product Range:**

expand inventory to offer a comprehensive selection of quality building materials and tools.
  3. **Customer Service:**

deliver friendly, knowledgeable, and personalized service to every customer.
  4. **Community Support:**

support local contractors and homeowners by providing reliable and timely product availability.
  5. **Business Growth:**

increase customer base and sales through consistent service and reputation building.

- Confirmation Letter

**To whom it may concern,**

I am writing to confirm that a team of undergraduates from NSBM has been granted approval to develop a Database System for **Jalitha Hardware** as part of their academic project. We are pleased to support their initiative and acknowledge their efforts in undertaking this valuable work.

The Database System proposed by the NSBM undergraduates represents a promising improvement to our business operations. It is expected to streamline our data management processes, enhance inventory control, and improve overall efficiency in handling customer and supplier records. We are optimistic about the positive changes this system will bring to our day-to-day activities.

We appreciate your role in facilitating this collaboration and look forward to fully supporting the students throughout the development of this system. The professionalism and dedication shown by the NSBM team are highly commendable, and we are confident that this project will be a success.

Thank you for your attention to this matter.

Sincerely,



Manager,  
Jalitha Hardware  
Chilaw Road,  
BandaraKoswaththa.

## Contents

Section 1.....	5
Scenario Overview.....	5
ER Diagram .....	6
Assumptions.....	8
Relational Mapping (ER to Relational Schema).....	10
Data Normalization.....	13
Data Dictionary .....	16
Section 2.....	20
Create Table .....	20
Database Diagram.....	31
Tables.....	32
Section 3.....	43
Trigger Statements.....	43
Function Statements.....	46
View Statements .....	48
Procedure Statements.....	51
Section 4.....	58
Hardware Management System Application .....	58
Application Drive Link .....	58
Application Features .....	58
Technologies Used.....	59
User Interface Design .....	60
Conclusion .....	63
Contribution Table.....	64

# Section 1

## Scenario Overview

Effective management of inventory, sales, and customer data is essential in contemporary retail hardware businesses. A hardware shop management system that tracks customers, employees, products, suppliers, sales orders, inventory, payments, and purchase orders is proposed in this project. The system ensures that employees can process sales transactions, customers can make purchases with flexible payment options, and suppliers can fulfill inventory requirements.

To achieve efficient data storage, retrieval, and integrity, the system emphasizes relational database design principles.

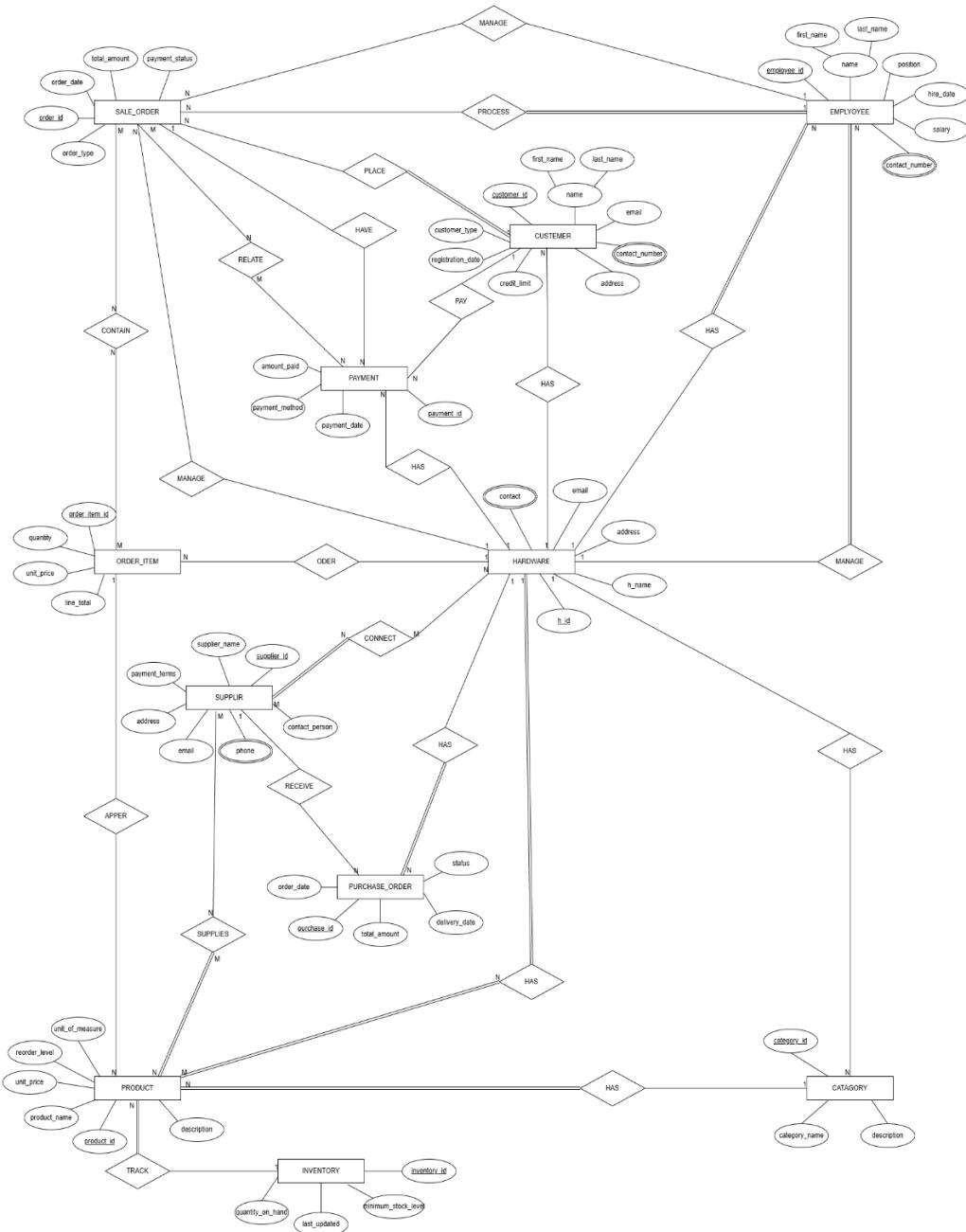
Jalitha Hardware operates as a single retail establishment specializing in construction materials, tools, electrical supplies, plumbing fixtures, and general hardware items. The business caters to both individual customers and contractors, managing high-volume daily transactions while maintaining accurate real-time inventory levels.

The proposed system must:

- Support multiple payment methods
- Generate professional invoices
- Handle credit sales
- Provide comprehensive reporting capabilities to assist in business decision-making

# ER Diagram

The following EER Diagram has all the main entities, attributes, and relationships.



## *Key Elements*

- Entities  
CUSTOMER, EMPLOYEE, PRODUCT, SUPPLIER, CATEGORY, SALES\_ORDER,  
ORDER\_ITEM, PAYMENT, INVENTORY, PURCHASE\_ORDER, HARDWARE
- Relationships  
MANAGE, PROCESS, PLACE, HAVE, RELATE, HAS, CONTAIN, ORDER, CONNECT, APPER,  
TRACK, SUPPLIES
- Cardinalities & Participation:
  - A customer can place multiple sales orders in the hardware shop system (1 : N).
  - A customer can make multiple payments in the hardware shop system (1 : N).
  - An employee can process multiple sales orders (1 : N)
  - Any sales order has an employee, and each employee processes many orders (1:N)
  - All products are supplied by suppliers in the system (N : 1)
  - The supplier supplies multiple products to the hardware shop (1 : N)
  - The category has many products (1 : N)
  - All products belong to specific categories (N : 1)
  - Sales orders contain multiple order items (1 : N)
  - Products can appear in multiple order items (1 : N)
  - Each product has specific inventory tracking (1 : 1)
  - All payments are related to specific sales orders (N : 1)
  - Each sales order can have multiple payments (1 : N)
  - Suppliers receive multiple purchase orders (1 : N)
  - Employees manage multiple purchase orders (1 : N)
  - Hardware has multiple products (1 :N)
  - Hardware has multiple categories of products(1 :N)
  - Hardware has a hardware management employee (1 :N)
  - Hardware has customers (N : 1)
  - Hardware manages sales orders and order items (1 :N)
  - Hardware purchase the ordered items (1 :N)
  - Hardware directly connects with suppliers ( N :M)
  - Hardware has a payment handling and management (N : 1)

## Assumptions

- The hardware store serves both private clients and contractors as a single shop.
- Multiple sales orders may be placed by the same customer, and one staff handles each order.
- Each employee is assigned to a single hardware store, and they are able to process multiple orders.
- Several providers supply items, and each supplier has the capacity to supply a large number of products.
- Every product falls into a specific category, and there are several products in each category.
- Each product can be found in many order items, and sales orders represent multiple products.
- Every product's inventory level is tracked separately, with real-time updates provided
- Each payment is linked to a single sales order, and customers can make several payments for their products.
- Orders for supplies are handled by the hardware store with suppliers, and one supplier may get several orders.
- The system creates detailed bills for each transaction and accepts an array of payment options.
- Each entity is used to number of attributes.
  - Customer - customer\_id, first\_name, last\_name, email, contact\_number, address, customer\_type, credit\_limit, registration\_date
  - Employee - employee\_id, first\_name, last\_name, position, hire\_date, salary, contact\_number
  - Product - product\_id, product\_name, brand, unit\_price, reorder\_level, unit\_of\_measure, description
  - Supplier - supplier\_id, supplier\_name, contact\_person, phone, email, address, payment\_terms
  - Category - category\_id, category\_name, description
  - Sales\_order - order\_id, order\_date, total\_amount, payment\_status, order\_type

- Order\_item - order\_item\_id, quantity, unit\_price, line\_total
- Payment - payment\_id, payment\_date, payment\_method, amount\_paid
- Inventory - inventory\_id, quantity\_on\_hand, last\_updated, minimum\_stock\_level
- Purchase\_order - purchase\_id, order\_date, total\_amount, delivery\_date, status
- Hardware - h\_id, h\_name, address, email, contact

The physical hardware store location is linked to all business processes, including sales, stock, and orders

## Relational Mapping (ER to Relational Schema)

Firstly, we are making our main database. The owner database name is hardware\_shop\_management\_system

- customer (customer\_id, first\_name, last\_name, email, contact\_number, address, customer\_type, credit\_limit, registration\_date)

customer_id	first_name	last_name	email	address
registration_date	contact_number	customer_type	credit_limit	customer_type
registration_date				

- employee (employee\_id, first\_name, last\_name, position, hire\_date, salary, contact\_number)

employee_id	first_name	last_name	position
hire_date	salary	contact_number	

- product (product\_id, product\_name, category\_id, supplier\_id, brand, unit\_price, reorder\_level, unit\_of\_measure, description)

product_id	product_name	category_id	supplier_id
brand	unit_price	reorder_level	unit_of_measure
description			

- supplier (supplier\_id, supplier\_name, contact\_person, phone, email, address, payment\_terms)

supplier_id	supplier_name	contact_person	phone
email	address	payment_terms	

- category (category\_id, category\_name, description)

category_id	category_name	description

- sales\_order (order\_id, customer\_id, employee\_id, order\_date, total\_amount, payment\_status, order\_type)

order_id	customer_id	employee_id	order_date	total_amount
payment_status	order_type			

- o order\_item (order\_item\_id, order\_id, product\_id, quantity, unit\_price, line\_total)

order_item_id	order_id	product_id	line_total	quantity
unit_price				

- o payment (payment\_id, order\_id, payment\_date, payment\_method, amount\_paid)

payment_id	order_id	payment_date	payment_method
amount_paid			

- o inventory (inventory\_id, product\_id, quantity\_on\_hand, last\_updated, minimum\_stock\_level)

inventory_id	product_id	quantity_on_hand	last_updated
minimum_stock_level			

- o purchase\_order (purchase\_id, supplier\_id, employee\_id, order\_date, total\_amount, delivery\_date, status)

purchase_id	supplier_id	employee_id	order_date	order_date
delivery_date	status			

- o hardware (h\_id, h\_name, address, email, contact, employee\_id, order\_id, payment\_id, supplier\_id, purchase\_id, category\_id)

h_id	h_name	address	email
contact	employee_id	order_id	payment_id
supplier_id	purchase_id	category_id	

- o hardware\_product (supplier\_id, product\_id)

supplier_id	product_id
-------------	------------

- supplier\_ hardware (supplier\_id, h\_id)

supplier_id	h_id
-------------	------

- product.hardware (h\_id, product\_id)

h_id	product_id
------	------------

- sales\_order\_order\_item (order\_id, order\_item\_id)

order_id	order_item_id
----------	---------------

- sales\_order\_payment (payment\_id, order\_id)

payment_id	order_id
------------	----------

## Data Normalization

- Final Normalized Tables (3NF)

- CUSTOMER (customer\_id, first\_name, last\_name, email, address, customer\_type, registration\_date)

customer_id	first_name	last_name	email	address
customer_type	registration_date			

- customer\_credit (credit\_limit, customer\_id, order\_id, payment\_id, h\_id)

credit_limit	customer_id	order_id	payment_id	h_id
--------------	-------------	----------	------------	------

- customer\_contact\_number (customer\_id, contact\_number)

customer_id	contact_number
-------------	----------------

- EMPLOYEE (employee\_id, first\_name, last\_name, position, hire\_date, salary, h\_id)

employee_id	first_name	last_name	position	hire_date
salary	h_id			

- employee\_contact\_number (employee\_id, contact\_number)

employee_id	contact_number
-------------	----------------

- PRODUCT (product\_id, product\_name, brand, unit\_price, reorder\_level, unit\_of\_measure, description, order\_item\_id, inventory\_id, category\_id)

product_id	product_name	brand	unit_price	reorder_level
unit_of_measure	description	order_item_id	inventory_id	category_id

- SUPPLIER (supplier\_id, supplier\_name, contact\_person, email, address)

supplier_id	supplier_name	contact_person	email	address
-------------	---------------	----------------	-------	---------

- supplier\_phone (supplier\_id, phone)

supplier_id	phone
-------------	-------

- supplier\_payment\_terms (supplier\_id, payment\_terms)

supplier_id	payment_terms
-------------	---------------

- CATEGORY (category\_id, category\_name, description, h\_id)

category_id	category_name	description	h_id
-------------	---------------	-------------	------

- SALES\_ORDER (order\_id, order\_date, total\_amount, payment\_status, order\_type, employee\_id, h\_id)

order_id	order_date	total_amount	payment_status
order_type	employee_id	h_id	

- ORDER\_ITEM (order\_item\_id, quantity, unit\_price, line\_total, h\_id)

order_item_id	quantity	unit_price	line_total	h_id
---------------	----------	------------	------------	------

- PAYMENT (payment\_id, payment\_date, payment\_method, amount\_paid, h\_id, order\_id)

payment_id	payment_date	payment_method	amount_paid
h_id	order_id		

- INVENTORY (inventory\_id, quantity\_on\_hand, last\_updated, minimum\_stock\_level)

inventory_id	quantity_on_hand	last_updated	minimum_stock_level
--------------	------------------	--------------	---------------------

- PURCHASES\_ORDER (purchase\_id, total\_amount, status, supplier\_id, h\_id)

purchase_id	total_amount	status	supplier_id	h_id
-------------	--------------	--------	-------------	------

- purchases\_date (purchase\_id, order\_date, delivery\_date)

purchase_id	order_date	delivery_date
-------------	------------	---------------

- HARDWARE (h\_id, h\_name, address, email)

h_id	h_name	address	email
------	--------	---------	-------

- hardware\_contact (h\_id, contact)

h_id	contact
------	---------

- hardware\_product (supplier\_id, product\_id)

supplier_id	product_id
-------------	------------

- supplier\_hardware (supplier\_id, h\_id)

supplier_id	h_id
-------------	------

- product.hardware (h\_id, product\_id)

h_id	product_id
------	------------

- sales\_order\_and\_item (order\_id, order\_item\_id)

order_id	order_item_id
----------	---------------

- sales\_order\_payment (payment\_id, order\_id)

payment_id	order_id
------------	----------

## Data Dictionary

- The updated Data Dictionary now includes a "Usage" column for each field, providing clarity on how each field is utilized in your University Course Management System.

table	attribute	Data type	constrain	usage
CUSTOMER	customer_id	INT	PRIMARY_KEY	Unique identifier for each customer
	first_name	VARCHAR	-	-
	last_name	VARCHAR	-	-
	email	UNIQUE	-	-
	address	TEXT	-	-
	customer_type	VARCHAR	-	-
	registration_date	DATE	-	-
customer_credit	order_id	INT	FOREAN_KEY	Linked to sale_order table
	customer_id	INT	FOREAN_KEY	Linked to customer table
	credit_limit,	DECIMAL	-	-
	payment_id	INT	FOREAN_KEY	Linked to payment table
	h_id	INT	FOREAN_KEY	Linked to hardware table
customer_contact_number	customer_id	INT	FOREAN_KEY	Linked to customer table
	contact_number	VARCHAR	--	-
EMPLOYEE	employee_id	INT	PRIMARY_KEY	Unique identifier for each employee
	first_name	VARCHAR	-	-
	last_name	VARCHAR	-	-
	position	VARCHAR	-	-
	hire_date	DATE	-	-
	salary	DECIMAL	--	-

employee_contact_number	employee_id	INT	FOREAN_KEY	Linked to employee table
	contact_number	VARCHARE	-	-
PRODUCT	product_id	INT	PRIMARY_KEY	Unique identifier for each product
	description	TEXT	-	-
	product_name	VARCHAR	-	-
	brand	VARCHAR	-	-
	unit_price	DECIMAL	-	-
	reorder_level	INT	-	-
	unit_of_measure	VARCHAR	-	-
	category_id	INT	FOREAN_KEY	Linked to category table
	order_item_id	INT	FOREAN_KEY	Linked to order_item table
SUPPLIER	inventory_id	INT	FOREAN_KEY	Linked to inventory table
	supplier_id	INT	PRIMARY_KEY	Unique identifier for each supplier
	supplier_name	VARCHAR	-	-
	address	VARCHAR	-	-
	email	UNIQUE	-	-
supplier_phone	contact_person	VARCHAR	-	-
	supplier_id	INT	FOREAN_KEY	Linked to supplier table
	phone	VARCHAR	-	-
supplier_payment_terms	supplier_id	INT	FOREAN_KEY	Linked to supplier table
	payment_terms	VARCHAR	--	-
CATEGORY	category_id	INT	PRIMARY_KEY	Unique identifier for each category
	category_name	VARCHAR	-	-
	description	TEXT	-	-
	h_id	INT	FOREAN_KEY	Linked to hardware table

SALES_ORDER	order_id	INT	PRIMARY_KEY	Unique identifier for each sales_order
	order_date	DATE	-	
	total_amount	DECIMAL	-	-
	payment_status	VARCHAR	-	-
	order_type	VARCHAR	-	-
	employee_id	INT	FOREAN_KEY	Linked to employee table
	customer_id	INT	FOREAN_KEY	Linked to customer table
ORDER_ITEM	order_item_id	INT	PRIMARY_KEY	Unique identifier for each order_item
	quantity	INT	-	-
	unit_price	DECIMAL	-	-
	line_total	DECIMAL	-	-
	h_id	INT	FOREAN_KEY	Linked to hardware table
PAYMENT	payment_id	INT	PRIMARY_KEY	Unique identifier for each payment
	payment_date	DATA	-	-
	payment_method	VARCHAR	-	-
	amount_paid	DECIMAL	-	-
	h_id	INT	FOREAN_KEY	-
	order_id	INT	FOREAN_KEY	-
INVENTORY	inventory_id	INT	PRIMARY_KEY	Unique identifier for each inentory
	quantity_on_hand	INT	-	-
	last_updated	DATE	-	-
	minimum_stock_level	INT	-	-
PURCHES_ORDER	purchase_id	INT	PRIMARY_KEY	Unique identifier for each purches_order
	total_amount	DECIMAL	-	-

	status	VARCHAR	-	-
	supplier_id	INT	FOREAN_KEY	Linked to supplier table
	h_id	INT	FOREAN_KEY	Linked to hardware table
purchase_date	purchase_id	INT	FOREAN_KEY	Linked to purchase table
	order_date	DATE	-	-
	order_date	DATE	-	-
HARDWARE	h_id	INT	PRIMARY_KEY	Unique identifier for each hardware
	h_name	VARCHAR	-	-
	address	VARCHAR	-	-
	email	UNIQUE	-	-
hardware_contact	h_id	INT	FOREAN_KEY	Linked to hardware table
	contact	VARCHAR		-
hardware_product	supplier_id	INT	FOREAN_KEY	Linked to supplier table
	product_id	INT	FOREAN_KEY	Linked to product table
supplier_hardware	supplier_id	INT	FOREAN_KEY	Linked to supplier table
	h_id	INT	FOREAN_KEY	Linked to hardware table
product.hardware	h_id	INT	FOREAN_KEY	Linked to hardware table
	product_id	INT	FOREAN_KEY	Linked to product table
sales_order_and_item	order_id	INT	FOREAN_KEY	Linked to sale_order table
	order_item_id	INT	FOREAN_KEY	Linked to order_item table
sales_order_payment	payment_id	INT	FOREAN_KEY	Linked to payment table
	order_id	INT	FOREAN_KEY	Linked to sale_order table

## Section 2

### Create Table Statements

#### 1. Hardware Table (Main Entity)

```
-- 1. HARDWARE Table (Main Entity)
CREATE TABLE HARDWARE (
    h_id INT PRIMARY KEY IDENTITY(1,1),
    h_name NVARCHAR(100) NOT NULL,
    address NVARCHAR(255) NOT NULL,
    email NVARCHAR(100) UNIQUE NOT NULL,
    CONSTRAINT CHK_Hardware_Email CHECK (email LIKE '%@%.%')
);
```

#### 2. Hardware Contact Table

```
-- 2. HARDWARE_CONTACT Table
CREATE TABLE HARDWARE_CONTACT (
    d_id INT PRIMARY KEY IDENTITY(1,1),
    h_id INT NOT NULL,
    contact NVARCHAR(15) NOT NULL,
    CONSTRAINT FK_HardwareContact_Hardware FOREIGN KEY (h_id)
        REFERENCES HARDWARE(h_id) ON DELETE CASCADE,
    CONSTRAINT CHK_Contact_Format CHECK (contact LIKE
        '[0-9]%' AND LEN(contact) >= 10)
);
```

### 3. Customer Table

```
-- 3. CUSTOMER Table
CREATE TABLE CUSTOMER (
    customer_id INT PRIMARY KEY IDENTITY(1,1),
    first_name NVARCHAR(50) NOT NULL,
    last_name NVARCHAR(50) NOT NULL,
    email NVARCHAR(100) UNIQUE NOT NULL,
    address NVARCHAR(255) NOT NULL,
    customer_type NVARCHAR(20) NOT NULL DEFAULT 'Individual',
    registration_date DATE NOT NULL DEFAULT GETDATE(),
    CONSTRAINT CHK_Customer_Type CHECK (customer_type IN
    ('Individual', 'Contractor', 'Business')),
    CONSTRAINT CHK_Customer_Email CHECK (email LIKE '%@%.%')
);
```

### 4. Customer Contact Number Table

```
-- 4. CUSTOMER_CONTACT_NUMBER Table
CREATE TABLE CUSTOMER_CONTACT_NUMBER (
    customer_id INT NOT NULL,
    contact_number NVARCHAR(15) NOT NULL,
    PRIMARY KEY (customer_id, contact_number),
    CONSTRAINT FK_CustomerContact_Customer FOREIGN KEY (customer_id)
    REFERENCES CUSTOMER(customer_id) ON DELETE CASCADE,
    CONSTRAINT CHK_CustomerContact_Format CHECK (contact_number LIKE
    '[0-9]%' AND LEN(contact_number) >= 10)
);
```

## 5. Employee Table

```
-- 5. EMPLOYEE Table
CREATE TABLE EMPLOYEE (
    employee_id INT PRIMARY KEY IDENTITY(1,1),
    first_name NVARCHAR(50) NOT NULL,
    last_name NVARCHAR(50) NOT NULL,
    position NVARCHAR(50) NOT NULL,
    hire_date DATE NOT NULL DEFAULT GETDATE(),
    salary DECIMAL(10,2) NOT NULL,
    CONSTRAINT CHK_Employee_Salary CHECK (salary > 0)
);
```

## 6. Employee Contact Number Table

```
-- 6. EMPLOYEE_CONTACT_NUMBER Table
CREATE TABLE EMPLOYEE_CONTACT_NUMBER (
    employee_id INT NOT NULL,
    contact_number NVARCHAR(15) NOT NULL,
    h_id INT NOT NULL,
    PRIMARY KEY (employee_id, contact_number),
    CONSTRAINT FK_EmployeeContact_Employee FOREIGN KEY
        (employee_id) REFERENCES EMPLOYEE(employee_id) ON DELETE CASCADE,
    CONSTRAINT FK_EmployeeContact_Hardware FOREIGN KEY
        (h_id) REFERENCES HARDWARE(h_id),
    CONSTRAINT CHK_EmployeeContact_Format CHECK (contact_number
        LIKE '[0-9]%' AND LEN(contact_number) >= 10)
);
```

## 7. Category Table

```
-- 7. CATEGORY Table
CREATE TABLE CATEGORY (
    category_id INT PRIMARY KEY IDENTITY(1,1),
    category_name NVARCHAR(50) NOT NULL UNIQUE,
    description NVARCHAR(255),
    h_id INT NOT NULL,
    CONSTRAINT FK_Category_Hardware FOREIGN KEY
        (h_id) REFERENCES HARDWARE(h_id)
);
```

## 8. Supplier Table

```
-- 8. SUPPLIER Table
CREATE TABLE SUPPLIER (
    supplier_id INT PRIMARY KEY IDENTITY(1,1),
    supplier_name NVARCHAR(100) NOT NULL,
    contact_person NVARCHAR(100) NOT NULL,
    email NVARCHAR(100) UNIQUE NOT NULL,
    address NVARCHAR(255) NOT NULL,
    CONSTRAINT CHK_Supplier_Email CHECK (email LIKE '%@%.%')
);
```

## 9. Supplier Phone Table

```
-- 9. SUPPLIER_PHONE Table
CREATE TABLE SUPPLIER_PHONE (
    supplier_id INT NOT NULL,
    phone NVARCHAR(15) NOT NULL,
    PRIMARY KEY (supplier_id, phone),
    CONSTRAINT FK_SupplierPhone_Supplier FOREIGN KEY
        (supplier_id) REFERENCES SUPPLIER(supplier_id)
    ON DELETE CASCADE,
    CONSTRAINT CHK_SupplierPhone_Format CHECK
        (phone LIKE '[0-9]%' AND LEN(phone) >= 10)
);
```

## 10. Supplier Payment Table

```
-- 10. SUPPLIER_PAYMENT_TERMS Table
CREATE TABLE SUPPLIER_PAYMENT_TERMS (
    supplier_id INT NOT NULL,
    payment_terms NVARCHAR(100) NOT NULL,
    PRIMARY KEY (supplier_id, payment_terms),
    CONSTRAINT FK_SupplierTerms_Supplier FOREIGN KEY
        (supplier_id) REFERENCES SUPPLIER(supplier_id) ON DELETE CASCADE
);
```

## 11. Inventory Table

```
-- 11. INVENTORY Table
CREATE TABLE INVENTORY (
    inventory_id INT PRIMARY KEY IDENTITY(1,1),
    quantity_on_hand INT NOT NULL DEFAULT 0,
    last_updated DATETIME NOT NULL DEFAULT GETDATE(),
    minimum_stock_level INT NOT NULL DEFAULT 0,
    CONSTRAINT CHK_Inventory_Quantity CHECK (quantity_on_hand >= 0),
    CONSTRAINT CHK_Inventory_MinStock CHECK (minimum_stock_level >= 0)
);
```

## 12.Order Item Table

```
-- 12. ORDER_ITEM Table
CREATE TABLE ORDER_ITEM (
    order_item_id INT PRIMARY KEY IDENTITY(1,1),
    quantity INT NOT NULL,
    unit_price DECIMAL(10,2) NOT NULL,
    line_total DECIMAL(10,2) NOT NULL,
    h_id INT NOT NULL,
    CONSTRAINT FK_OrderItem_Hardware FOREIGN KEY (h_id) REFERENCES HARDWARE(h_id),
    CONSTRAINT CHK_OrderItem_Quantity CHECK (quantity > 0),
    CONSTRAINT CHK_OrderItem_Price CHECK (unit_price > 0),
    CONSTRAINT CHK_OrderItem_Total CHECK (line_total > 0)
);
```

## 13.Product Table

```
-- 13. PRODUCT Table
CREATE TABLE PRODUCT (
    product_id INT PRIMARY KEY IDENTITY(1,1),
    product_name NVARCHAR(100) NOT NULL,
    brand NVARCHAR(50),
    unit_price DECIMAL(10,2) NOT NULL,
    reorder_level INT NOT NULL DEFAULT 10,
    unit_of_measure NVARCHAR(20) NOT NULL DEFAULT 'PCS',
    description NVARCHAR(255),
    order_item_id INT,
    inventory_id INT NOT NULL,
    category_id INT NOT NULL,
    CONSTRAINT FK_Product_OrderItem FOREIGN KEY (order_item_id)
        REFERENCES ORDER_ITEM(order_item_id),
    CONSTRAINT FK_Product_Inventory FOREIGN KEY (inventory_id)
        REFERENCES INVENTORY(inventory_id),
    CONSTRAINT FK_Product_Category FOREIGN KEY (category_id)
        REFERENCES CATEGORY(category_id),
    CONSTRAINT CHK_Product_Price CHECK (unit_price > 0),
    CONSTRAINT CHK_Product_ReorderLevel CHECK (reorder_level >= 0)
);
```

#### 14.Sales Order Table

```
-- 14. SALES_ORDER Table
CREATE TABLE SALES_ORDER (
    order_id INT PRIMARY KEY IDENTITY(1,1),
    order_date DATE NOT NULL DEFAULT GETDATE(),
    total_amount DECIMAL(12,2) NOT NULL,
    payment_status NVARCHAR(20) NOT NULL DEFAULT 'Pending',
    order_type NVARCHAR(20) NOT NULL DEFAULT 'Sale',
    employee_id INT NOT NULL,
    h_id INT NOT NULL,
    CONSTRAINT FK_SalesOrder_Employee FOREIGN KEY (employee_id)
    REFERENCES EMPLOYEE(employee_id),
    CONSTRAINT FK_SalesOrder_Hardware FOREIGN KEY (h_id)
    REFERENCES HARDWARE(h_id),
    CONSTRAINT CHK_SalesOrder_Amount CHECK (total_amount > 0),
    CONSTRAINT CHK_Payment_Status CHECK (payment_status IN
    ('Pending', 'Partial', 'Paid', 'Cancelled')),
    CONSTRAINT CHK_Order_Type CHECK (order_type IN
    ('Sale', 'Return', 'Exchange'))
);
```

#### 15.Payment Table

```
-- 15. PAYMENT Table
CREATE TABLE PAYMENT (
    payment_id INT PRIMARY KEY IDENTITY(1,1),
    payment_date DATE NOT NULL DEFAULT GETDATE(),
    payment_method NVARCHAR(20) NOT NULL,
    amount_paid DECIMAL(10,2) NOT NULL,
    h_id INT NOT NULL,
    order_id INT NOT NULL,
    CONSTRAINT FK_Payment_Hardware FOREIGN KEY (h_id)
    REFERENCES HARDWARE(h_id),
    CONSTRAINT FK_Payment_SalesOrder FOREIGN KEY (order_id)
    REFERENCES SALES_ORDER(order_id),
    CONSTRAINT CHK_Payment_Amount CHECK (amount_paid > 0),
    CONSTRAINT CHK_Payment_Method CHECK (payment_method IN
    ('Cash', 'Credit Card', 'Debit Card', 'Bank Transfer', 'Check'))
);
```

## 16.Customer Credit Table

```
-- 16. CUSTOMER_CREDIT Table
CREATE TABLE CUSTOMER_CREDIT (
    customer_id INT NOT NULL,
    order_id INT NOT NULL,
    payment_id INT NOT NULL,
    h_id INT NOT NULL,
    credit_limit DECIMAL(10,2) NOT NULL DEFAULT 0,
    PRIMARY KEY (customer_id, order_id, payment_id),
    CONSTRAINT FK_CustomerCredit_Customer FOREIGN KEY
    (customer_id) REFERENCES CUSTOMER(customer_id),
    CONSTRAINT FK_CustomerCredit_Order FOREIGN KEY
    (order_id) REFERENCES SALES_ORDER(order_id),
    CONSTRAINT FK_CustomerCredit_Payment FOREIGN KEY
    (payment_id) REFERENCES PAYMENT(payment_id),
    CONSTRAINT FK_CustomerCredit_Hardware FOREIGN KEY
    (h_id) REFERENCES HARDWARE(h_id),
    CONSTRAINT CHK_Credit_Limit CHECK (credit_limit >= 0)
);
```

## 17.Purchase Order Table

```
-- 17. PURCHASE_ORDER Table
CREATE TABLE PURCHASE_ORDER (
    purchase_id INT PRIMARY KEY IDENTITY(1,1),
    total_amount DECIMAL(12,2) NOT NULL,
    status NVARCHAR(20) NOT NULL DEFAULT 'Pending',
    supplier_id INT NOT NULL,
    h_id INT NOT NULL,
    CONSTRAINT FK_PurchaseOrder_Supplier FOREIGN KEY
    (supplier_id) REFERENCES SUPPLIER(supplier_id),
    CONSTRAINT FK_PurchaseOrder_Hardware FOREIGN KEY
    (h_id) REFERENCES HARDWARE(h_id),
    CONSTRAINT CHK_PurchaseOrder_Amount CHECK (total_amount > 0),
    CONSTRAINT CHK_Purchase_Status CHECK (status IN
    ('Pending', 'Approved', 'Shipped', 'Delivered', 'Cancelled'))
);
```

## 18.Purchase Date Table

```
-- 18. PURCHASE_DATE Table |
CREATE TABLE PURCHASE_DATE (
    purchase_id INT NOT NULL,
    order_date DATE NOT NULL DEFAULT GETDATE(),
    delivery_date DATE,
    PRIMARY KEY (purchase_id, order_date),
    CONSTRAINT FK_PurchaseDate_PurchaseOrder
    FOREIGN KEY (purchase_id) REFERENCES PURCHASE_ORDER
    (purchase_id) ON DELETE CASCADE,
    CONSTRAINT CHK_Delivery_Date CHECK (delivery_date
    IS NULL OR delivery_date >= order_date)
);
```

## 19.Hardware Product Table

```
-- 19. HARDWARE_PRODUCT Table
CREATE TABLE HARDWARE_PRODUCT (
    h_id INT NOT NULL,
    product_id INT NOT NULL,
    PRIMARY KEY (h_id, product_id),
    CONSTRAINT FK_HardwareProduct_Hardware FOREIGN KEY
    (h_id) REFERENCES HARDWARE(h_id) ON DELETE CASCADE,
    CONSTRAINT FK_HardwareProduct_Product FOREIGN KEY
    (product_id) REFERENCES PRODUCT(product_id) ON DELETE CASCADE
);
```

## 20. Supplier Hardware Table

```
-- 20. SUPPLIER_HARDWARE Table
CREATE TABLE SUPPLIER_HARDWARE (
    supplier_id INT NOT NULL,
    h_id INT NOT NULL,
    PRIMARY KEY (supplier_id, h_id),
    CONSTRAINT FK_SupplierHardware_Supplier FOREIGN KEY
        (supplier_id) REFERENCES SUPPLIER(supplier_id) ON DELETE CASCADE,
    CONSTRAINT FK_SupplierHardware_Hardware FOREIGN KEY
        (h_id) REFERENCES HARDWARE(h_id) ON DELETE CASCADE
);
```

## 21. Product Hardware Table

```
-- 21. PRODUCT_HARDWARE Table
CREATE TABLE PRODUCT_HARDWARE (
    h_id INT NOT NULL,
    product_id INT NOT NULL,
    PRIMARY KEY (h_id, product_id),
    CONSTRAINT FK_ProductHardware_Hardware FOREIGN KEY
        (h_id) REFERENCES HARDWARE(h_id) ON DELETE CASCADE,
    CONSTRAINT FK_ProductHardware_Product FOREIGN KEY
        (product_id) REFERENCES PRODUCT(product_id) ON DELETE CASCADE
);
```

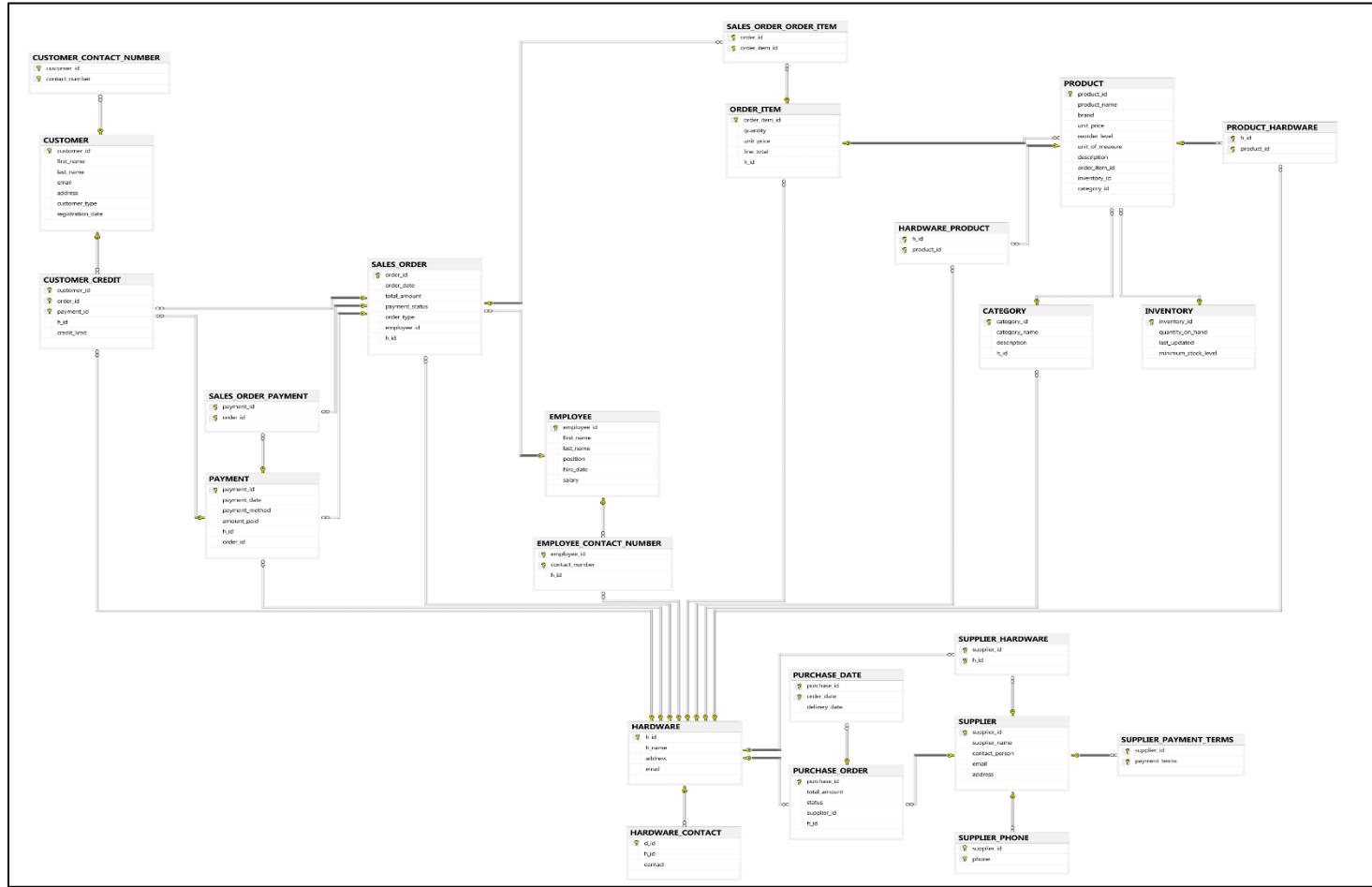
## 22. Sales Order Item Table

```
-- 22. SALES_ORDER_ORDER_ITEM Table
CREATE TABLE SALES_ORDER_ORDER_ITEM (
    order_id INT NOT NULL,
    order_item_id INT NOT NULL,
    PRIMARY KEY (order_id, order_item_id),
    CONSTRAINT FK_SalesOrderItem_SalesOrder FOREIGN KEY
        (order_id) REFERENCES SALES_ORDER(order_id) ON DELETE CASCADE,
    CONSTRAINT FK_SalesOrderItem_OrderItem FOREIGN KEY
        (order_item_id) REFERENCES ORDER_ITEM(order_item_id) ON DELETE CASCADE
);
```

## 23.Sales Order Payment Table

```
-- 23. SALES_ORDER_PAYMENT Table
CREATE TABLE SALES_ORDER_PAYMENT (
    payment_id INT NOT NULL,
    order_id INT NOT NULL,
    PRIMARY KEY (payment_id, order_id),
    CONSTRAINT FK_SalesOrderPayment_Payment FOREIGN KEY
    (payment_id) REFERENCES PAYMENT(payment_id) ON DELETE CASCADE,
    CONSTRAINT FK_SalesOrderPayment_SalesOrder FOREIGN KEY
    (order_id) REFERENCES SALES_ORDER(order_id) ON DELETE CASCADE
);
```

# Database Diagram



## Tables

### 1. HARDWARE data

	h_id	h_name	address	email
1	1	Jalitha Hardware	Bandarakoswaththa, Kurunegala	info@jalithahardware.lk

### 2. HARDWARE\_CONTACT data

	d_id	h_id	contact
1	1	1	0112345678
2	2	1	0777123456

### 3. CUSTOMER data

	customer_id	first_name	last_name	email	address	customer_type	registration_date
1	1	Amara	Silva	amara.silva@email.com	12 Temple Road, Colombo 03	Individual	2024-01-15
2	2	Kasun	Perera	kasun.perera@email.com	45 Lake Drive, Kandy	Contractor	2024-01-20
3	3	Nimal	Fernando	nimal.fernando@email.com	78 Beach Road, Negombo	Business	2024-02-05
4	4	Sunitha	Jayawardena	sunitha.j@email.com	23 Hill Street, Nuwara Eliya	Individual	2024-02-10
5	5	Ruwan	Wickramasinghe	ruwan.w@email.com	67 Main Road, Galle	Contractor	2024-02-15
6	6	Chamara	Rajapaksa	chamara.r@email.com	89 Station Road, Matara	Individual	2024-03-01
7	7	Dilini	Gunawardena	dilini.g@email.com	34 Park Lane, Colombo 07	Business	2024-03-05
8	8	Pradeep	Amarasinghe	pradeep.a@email.com	56 School Lane, Kurunegala	Individual	2024-03-10
9	9	Malini	Dissanayake	malini.d@email.com	78 Church Street, Badulla	Contractor	2024-03-15
10	10	Saman	Rathnayake	saman.r@email.com	90 Market Street, Anuradhapura	Individual	2024-03-20
11	11	Thilani	Mendis	thilani.m@email.com	12 River Road, Polonnaruwa	Business	2024-04-01
12	12	Ajith	Bandara	ajith.b@email.com	45 Forest Lane, Hambantota	Contractor	2024-04-05

#### **4. CUSTOMER\_CONTACT\_NUMBER data**

	customer_id	contact_number
1	1	0112345678
2	1	0771234567
3	2	0772345678
4	2	0812234567
5	3	0312234567
6	3	0773456789
7	4	0522234567
8	4	0774567890
9	5	0775678901
10	5	0912234567
11	6	0412234567
12	6	0776789012
13	7	0112345679
14	7	0777890123
15	8	0372234567
16	8	0778901234
17	9	0552234567
18	9	0779012345
19	10	0252234567
20	10	0770123456

#### **5. EMPLOYEE data**

	employee_id	first_name	last_name	position	hire_date	salary
1	1	Rajitha	Perera	Manager	2023-01-15	85000.00
2	2	Kumari	Silva	Sales Assistant	2023-02-01	45000.00
3	3	Dinesh	Femando	Store Keeper	2023-03-10	50000.00
4	4	Sandya	Jayawardena	Cashier	2023-04-05	40000.00
5	5	Priyantha	Wickramasinghe	Sales Executive	2023-05-20	55000.00
6	6	Nayana	Rajapaksa	Inventory Manager	2023-06-15	70000.00
7	7	Asanka	Gunawardena	Delivery Driver	2023-07-01	35000.00
8	8	Shirani	Amarasinghe	Customer Service	2023-08-10	42000.00
9	9	Chathura	Dissanayake	Purchase Officer	2023-09-05	60000.00
10	10	Ruvini	Rathnayake	Accountant	2023-10-20	65000.00
11	11	Janaka	Mendis	Security Officer	2023-11-15	30000.00
12	12	Darshani	Bandara	Receptionist	2023-12-01	38000.00

## 6. EMPLOYEE\_CONTACT\_NUMBER data

	employee_id	contact_number	h_id
1	1	0112111111	1
2	1	0771111111	1
3	2	0112222222	1
4	2	0772222222	1
5	3	0112333333	1
6	3	0773333333	1
7	4	0112444444	1
8	4	0774444444	1
9	5	0112555555	1
10	5	0775555555	1
11	6	0112666666	1

## 7. CATEGORY data

	category_id	category_name	description	h_id
1	1	Hand Tools	Hammers, screwdrivers, wrenches, and other manual tools	1
2	2	Power Tools	Electric drills, saws, grinders, and power equipment	1
3	3	Building Materials	Cement, sand, bricks, and construction materials	1
4	4	Electrical Supplies	Wires, switches, outlets, and electrical components	1
5	5	Plumbing Fixtures	Pipes, fittings, valves, and plumbing supplies	1
6	6	Hardware Fasteners	Screws, bolts, nuts, washers, and fastening hardware	1
7	7	Paint & Finishes	Paints, brushes, rollers, and finishing materials	1
8	8	Garden Tools	Shovels, rakes, hoses, and gardening equipment	1
9	9	Safety Equipment	Helmets, gloves, goggles, and protective gear	1
10	10	Measuring Tools	Rulers, levels, measuring tapes, and precision instruments	1
11	11	Adhesives & Sealants	Glues, tapes, caulk, and sealing materials	1
12	12	Cleaning Supplies	Detergents, brushes, and maintenance materials	1

## 8. SUPPLIER data

supplier_id	supplier_name	contact_person	email	address
1	ABC Hardware Distributors	John Perera	john@abchardware.lk	100 Industrial Zone, Colombo 15
2	XYZ Building Supplies	Mary Silva	mary@xyzbuild.lk	200 Warehouse Road, Kelaniya
3	Prime Tools Lanka	David Fernando	david@primetools.lk	300 Factory Lane, Ja-Ela
4	Elite Electrical Co.	Sarah Jayawardena	sarah@eliteelectrical.lk	400 Electric Avenue, Nugegoda
5	Superior Plumbing Ltd.	Michael Wickrama	michael@superiorplumb.lk	500 Pipe Street, Dehiwala
6	Universal Fasteners	Lisa Rajapaksa	lisa@unifast.lk	600 Bolt Boulevard, Moratuwa
7	Quality Paint Solutions	Robert Gunawardena	robert@qualitypaint.lk	700 Color Road, Maharagama
8	Green Garden Tools	Jennifer Amarasinghe	jennifer@greengarden.lk	800 Garden Way, Kottawa
9	Safety First Lanka	William Dissanayake	william@safetyfirst.lk	900 Safety Street, Pannipitiya
10	Precision Measuring Co.	Amanda Rathnayake	amanda@precision.lk	1000 Measure Mile, Battaramulla
11	Bond Adhesives Ltd.	Steven Mendis	steven@bondadhesives.lk	1100 Stick Street, Rajagiriya
12	Clean Care Supplies	Michelle Bandara	micelle@cleancare.lk	1200 Clean Circle, Thalawathugoda

## 9. SUPPLIER\_PHONE data

supplier_id	phone
1	0112345001
1	0777345001
2	0112345002
2	0777345002
3	0112345003
3	0777345003
4	0112345004
4	0777345004
5	0112345005
5	0777345005
6	0112345006
6	0777345006
7	0112345007
7	0777345007
8	0112345008
8	0777345008
9	0112345009
9	0777345009
10	0112345010

## 10.SUPPLIER\_PAYMENT\_TERMS data

supplier_id	payment_terms
1	Cash on Delivery
1	Net 30 Days
2	2% 10 Net 30
2	Net 15 Days
3	Cash on Delivery
3	Net 45 Days
4	Monthly Credit
4	Net 30 Days
5	Cash on Delivery
5	Net 21 Days
6	1% 15 Net 30
6	Net 30 Days
7	Cash on Delivery
7	Net 14 Days
8	Net 30 Days
8	Weekly Credit
9	Cash on Delivery
9	Net 60 Days
10	3% 10 Net 30
10	Net 30 Days

## 11.INVENTORY data

	inventory_id	quantity_on_hand	last_updated	minimum_stock_level
1	1	150	2024-12-01 09:00:00.000	20
2	2	75	2024-12-01 10:15:00.000	10
3	3	200	2024-12-01 11:30:00.000	25
4	4	45	2024-12-01 12:45:00.000	15
5	5	120	2024-12-01 14:00:00.000	30
6	6	300	2024-12-01 15:15:00.000	50
7	7	80	2024-12-01 16:30:00.000	12
8	8	60	2024-12-01 09:45:00.000	8
9	9	180	2024-12-01 10:30:00.000	25
10	10	95	2024-12-01 11:15:00.000	15
11	11	220	2024-12-01 13:00:00.000	40
12	12	110	2024-12-01 14:45:00.000	20

## 12.ORDER\_ITEM data

order_item_id	quantity	unit_price	line_total	h_id
1	5	1500.00	7500.00	1
2	2	3500.00	7000.00	1
3	10	250.00	2500.00	1
4	3	1200.00	3600.00	1
5	8	450.00	3600.00	1
6	1	8500.00	8500.00	1
7	6	180.00	1080.00	1
8	4	720.00	2880.00	1
9	12	95.00	1140.00	1
10	7	650.00	4550.00	1
11	15	85.00	1275.00	1
12	9	320.00	2880.00	1

## 13.PRODUCT data

product_id	product_name	brand	unit_price	reorder_level	unit_of_measure	description	order_item_id	inventory_id	category_id
1	Claw Hammer 16oz	Stanley	1500.00	20	PCS	Professional claw hammer with fiberglass handle	1	1	1
2	Cordless Drill 18V	Bosch	3500.00	10	PCS	Lithium-ion cordless drill with 2 batteries	2	2	2
3	Portland Cement 50kg	Holcim	250.00	100	BAG	High quality portland cement for construction	3	3	3
4	Circuit Breaker 20A	Siemens	1200.00	25	PCS	Single pole circuit breaker 20 ampere	4	4	4
5	PVC Pipe 4inch	Astral	450.00	50	MTR	High quality PVC pipe for plumbing	5	5	5
6	Angle Grinder 4.5inch	Makita	8500.00	5	PCS	Professional angle grinder with safety guard	6	6	2
7	Wood Screws 2inch	Generic	180.00	200	BOX	Phillips head wood screws pack of 100	7	7	6
8	Safety Helmet	MSA	720.00	30	PCS	Industrial safety helmet with chin strap	8	8	9
9	Wall Paint White 4L	Dulux	950.00	40	CAN	Premium quality interior wall paint	9	9	7
10	Garden Spade	Fiskars	650.00	15	PCS	Steel blade garden spade with wooden handle	10	10	8
11	Measuring Tape 5M	Stanley	850.00	25	PCS	Heavy duty measuring tape with magnetic tip	11	11	10
12	Super Glue 20g	3M	320.00	80	TUB	Instant bonding super glue for multiple surfaces	12	12	11

## 14.SALES\_ORDER data

order_id	order_date	total_amount	payment_status	order_type	employee_id	h_id
1	2024-12-01	17500.00	Paid	Sale	1	1
2	2024-12-01	9100.00	Paid	Sale	2	1
3	2024-12-02	12100.00	Partial	Sale	3	1
4	2024-12-02	5680.00	Paid	Sale	4	1
5	2024-12-03	21850.00	Pending	Sale	5	1
6	2024-12-03	8320.00	Paid	Sale	1	1
7	2024-12-04	15750.00	Paid	Sale	2	1
8	2024-12-04	6890.00	Partial	Sale	3	1
9	2024-12-05	19200.00	Paid	Sale	4	1
10	2024-12-05	11450.00	Pending	Sale	5	1
11	2024-12-06	7830.00	Paid	Sale	1	1
12	2024-12-06	13920.00	Paid	Sale	2	1

## 15.PAYMENT data

payment_id	payment_date	payment_method	amount_paid	h_id	order_id
1	2024-12-01	Cash	17500.00	1	1
2	2024-12-01	Credit Card	9100.00	1	2
3	2024-12-02	Cash	8000.00	1	3
4	2024-12-02	Debit Card	5680.00	1	4
5	2024-12-03	Bank Transfer	8320.00	1	6
6	2024-12-04	Cash	15750.00	1	7
7	2024-12-04	Credit Card	4000.00	1	8
8	2024-12-05	Cash	19200.00	1	9
9	2024-12-06	Debit Card	7830.00	1	11
10	2024-12-06	Cash	13920.00	1	12
11	2024-12-07	Credit Card	5500.00	1	3
12	2024-12-07	Bank Transfer	2890.00	1	8

## 16.CUSTOMER\_CREDIT data

	customer_id	order_id	payment_id	h_id	credit_limit
1	2	1	1	1	50000.00
2	2	2	2	1	50000.00
3	2	3	3	1	50000.00
4	2	12	10	1	50000.00
5	5	5	5	1	75000.00
6	5	6	5	1	75000.00
7	7	7	6	1	40000.00
8	7	8	7	1	40000.00
9	9	9	8	1	60000.00
10	9	10	8	1	60000.00
11	11	4	4	1	45000.00
12	11	11	9	1	45000.00

## 17.PURCHASE\_ORDER data

purchase_id	total_amount	status	supplier_id	h_id
1	125000.00	Delivered	1	1
2	89500.00	Shipped	2	1
3	156200.00	Approved	3	1
4	67800.00	Delivered	4	1
5	98700.00	Pending	5	1
6	134500.00	Shipped	6	1
7	76200.00	Delivered	7	1
8	112300.00	Approved	8	1
9	85600.00	Delivered	9	1
10	143800.00	Shipped	10	1
11	94500.00	Pending	11	1
12	118700.00	Delivered	12	1

## 18.PURCHASE\_DATE data

purchase_id	order_date	delivery_date
1	2024-11-15	2024-11-25
2	2024-11-18	2024-12-02
3	2024-11-20	NULL
4	2024-11-22	2024-11-30
5	2024-11-25	NULL
6	2024-11-28	2024-12-08
7	2024-12-01	2024-12-05
8	2024-12-03	NULL
9	2024-12-05	2024-12-10
10	2024-12-08	2024-12-18
11	2024-12-10	NULL
12	2024-12-12	2024-12-15

## 19.HARDWARE\_PRODUCT data

order_id	order_item_id
1	1
1	2
2	3
2	4
3	5
3	6
4	7
4	8
5	9
5	10
6	11
6	12
7	1
7	3
8	2
8	4
9	5
9	7
10	6
10	8

## 20.SUPPLIER\_HARDWARE data

supplier_id	h_id
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1

## 21.PRODUCT\_HARDWARE data

h_id	product_id
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10
1	11
1	12

## 22.SALES\_ORDER\_ORDER\_ITEM data

order_id	order_item_id
1	1
1	2
2	3
2	4
3	5
3	6
4	7
4	8
5	9
5	10
6	11
6	12
7	1
7	3
8	2
8	4
9	5
9	7
10	6
10	8

## **23.SALES\_ORDER\_PAYMENT data**

payment_id	order_id
1	1
2	2
3	3
4	4
5	6
6	7
7	8
8	9
9	11
10	12
11	3
12	8

## Section 3

### Trigger Statements

```
CREATE TRIGGER trg_UpdateInventoryAftersale
ON SALES_ORDER_ORDER_ITEM
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @product_id INT, @quantity_sold INT, @inventory_id INT;

    -- Get the product and quantity from the inserted order item
    SELECT @product_id = p.product_id, @quantity_sold = oi.quantity, @inventory_id = p.inventory_id
    FROM inserted i
    INNER JOIN ORDER_ITEM oi ON i.order_item_id = oi.order_item_id
    INNER JOIN PRODUCT p ON oi.order_item_id = p.order_item_id;

    -- Update inventory quantity
    UPDATE INVENTORY
    SET quantity_on_hand = quantity_on_hand - @quantity_sold,
        last_updated = GETDATE()
    WHERE inventory_id = @inventory_id;

    -- Check if inventory falls below minimum stock level
    IF EXISTS (SELECT 1 FROM INVENTORY WHERE inventory_id = @inventory_id
               AND quantity_on_hand < minimum_stock_level)
    BEGIN
        PRINT 'WARNING: Product ID ' + CAST(@product_id AS VARCHAR(10)) +
              ' has fallen below minimum stock level!';
    END
END;
```

```

CREATE TRIGGER trg_ValidatePaymentAmount
ON PAYMENT
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @order_id INT, @total_payments DECIMAL(12,2), @order_total DECIMAL(12,2);

    -- Check each affected order
    SELECT DISTINCT @order_id = order_id FROM inserted;

    -- Calculate total payments for this order
    SELECT @total_payments = ISNULL(SUM(amount_paid), 0)
    FROM PAYMENT
    WHERE order_id = @order_id;

    -- Get order total
    SELECT @order_total = total_amount
    FROM SALES_ORDER
    WHERE order_id = @order_id;

    -- Validate payment doesn't exceed order total
    IF @total_payments > @order_total
    BEGIN
        RAISERROR('Total payments cannot exceed order amount', 16, 1);
        ROLLBACK TRANSACTION;
    END

    -- Update payment status based on total payments
    UPDATE SALES_ORDER
    SET payment_status = CASE
        WHEN @total_payments = 0 THEN 'Pending'
        WHEN @total_payments < @order_total THEN 'Partial'
        WHEN @total_payments = @order_total THEN 'Paid'
        ELSE 'Overpaid'
    END
    WHERE order_id = @order_id;
END;

```

```

-- TRIGGER 3: Audit Trail for Product Price Changes
-- This trigger maintains an audit log of product price changes
CREATE TABLE PRODUCT_PRICE_AUDIT (
    audit_id INT IDENTITY(1,1) PRIMARY KEY,
    product_id INT,
    old_price DECIMAL(10,2),
    new_price DECIMAL(10,2),
    change_date DATETIME DEFAULT GETDATE(),
    changed_by NVARCHAR(100) DEFAULT SYSTEM_USER
);
GO

CREATE TRIGGER trg_ProductPriceAudit
ON PRODUCT
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Only log if unit_price actually changed
    IF UPDATE(unit_price)
    BEGIN
        INSERT INTO PRODUCT_PRICE_AUDIT (product_id, old_price, new_price)
        SELECT d.product_id, d.unit_price, i.unit_price
        FROM deleted d
        INNER JOIN inserted i ON d.product_id = i.product_id
        WHERE d.unit_price != i.unit_price;
    END
END;
GO

```

## Function Statements

```
-- 2. USER DEFINED FUNCTIONS
-- =====

-- FUNCTION 1: Calculate Customer Total Purchases
-- This function calculates the total amount a customer has purchased
CREATE FUNCTION fn_GetCustomerTotalPurchases(@customer_id INT)
RETURNS DECIMAL(12,2)
AS
BEGIN
    DECLARE @total DECIMAL(12,2);

    SELECT @total = ISNULL(SUM(so.total_amount), 0)
    FROM SALES_ORDER so
    INNER JOIN CUSTOMER_CREDIT cc ON so.order_id = cc.order_id
    WHERE cc.customer_id = @customer_id;

    RETURN @total;
END;

-- FUNCTION 2: Get Product Stock Status
-- This function returns the stock status of a product
CREATE FUNCTION fn_GetProductStockStatus(@product_id INT)
RETURNS NVARCHAR(20)
AS
BEGIN
    DECLARE @status NVARCHAR(20);
    DECLARE @current_stock INT, @min_stock INT;

    SELECT @current_stock = i.quantity_on_hand, @min_stock = i.minimum_stock_level
    FROM PRODUCT p
    INNER JOIN INVENTORY i ON p.inventory_id = i.inventory_id
    WHERE p.product_id = @product_id;

    SET @status = CASE
        WHEN @current_stock = 0 THEN 'Out of Stock'
        WHEN @current_stock <= @min_stock THEN 'Low Stock'
        WHEN @current_stock <= (@min_stock * 2) THEN 'Medium Stock'
        ELSE 'High Stock'
    END;

    RETURN @status;
END;
```

```

]-- FUNCTION 3: Calculate Monthly Sales Total
-- This function calculates total sales for a specific month and year
]CREATE FUNCTION fn_GetMonthlySales(@year INT, @month INT)
RETURNS DECIMAL(12,2)
AS
BEGIN
    DECLARE @total DECIMAL(12,2);

    SELECT @total = ISNULL(SUM(total_amount), 0)
    FROM SALES_ORDER
    WHERE YEAR(order_date) = @year
    AND MONTH(order_date) = @month
    AND payment_status != 'Cancelled';

    RETURN @total;
END;

]-- FUNCTION 4: Get Supplier Performance Rating
-- This function calculates supplier performance based on delivery times
]CREATE FUNCTION fn_GetSupplierPerformance(@supplier_id INT)
RETURNS NVARCHAR(20)
AS
BEGIN
    DECLARE @rating NVARCHAR(20);
    DECLARE @avg_delay FLOAT;

    SELECT @avg_delay = AVG(DATEDIFF(day, pd.order_date, pd.delivery_date))
    FROM PURCHASE_ORDER po
    INNER JOIN PURCHASE_DATE pd ON po.purchase_id = pd.purchase_id
    WHERE po.supplier_id = @supplier_id
    AND pd.delivery_date IS NOT NULL
    AND po.status = 'Delivered';

    SET @rating = CASE
        WHEN @avg_delay IS NULL THEN 'No Data'
        WHEN @avg_delay <= 0 THEN 'Excellent'
        WHEN @avg_delay <= 2 THEN 'Good'
        WHEN @avg_delay <= 5 THEN 'Average'
        ELSE 'Poor'
    END;

    RETURN @rating;
END;
--
```

## View Statements

```
-- VIEW 1: Customer Sales Summary
-- This view provides a comprehensive summary of customer sales information
CREATE VIEW vw_CustomerSalesSummary
AS
SELECT
    c.customer_id,
    c.first_name + ' ' + c.last_name AS customer_name,
    c.email,
    c.customer_type,
    COUNT(DISTINCT so.order_id) AS total_orders,
    ISNULL(SUM(so.total_amount), 0) AS total_purchases,
    ISNULL(AVG(so.total_amount), 0) AS average_order_value,
    MAX(so.order_date) AS last_order_date,
    CASE
        WHEN COUNT(DISTINCT so.order_id) = 0 THEN 'Inactive'
        WHEN MAX(so.order_date) >= DATEADD(month, -3, GETDATE()) THEN 'Active'
        WHEN MAX(so.order_date) >= DATEADD(month, -12, GETDATE()) THEN 'Occasional'
        ELSE 'Inactive'
    END AS customer_status
FROM CUSTOMER c
LEFT JOIN CUSTOMER_CREDIT cc ON c.customer_id = cc.customer_id
LEFT JOIN SALES_ORDER so ON cc.order_id = so.order_id
GROUP BY c.customer_id, c.first_name, c.last_name, c.email, c.customer_type;
GO
```

```

-- VIEW 2: Inventory Management Dashboard
-- This view provides current inventory status for all products
CREATE VIEW vw_InventoryDashboard
AS
SELECT
    p.product_id,
    p.product_name,
    p.brand,
    c.category_name,
    p.unit_price,
    i.quantity_on_hand,
    i.minimum_stock_level,
    i.last_updated,
    s.supplier_name,
    CASE
        WHEN i.quantity_on_hand = 0 THEN 'Out of Stock'
        WHEN i.quantity_on_hand <= i.minimum_stock_level THEN 'Low Stock'
        WHEN i.quantity_on_hand <= (i.minimum_stock_level * 2) THEN 'Medium Stock'
        ELSE 'High Stock'
    END AS stock_status,
    (p.unit_price * i.quantity_on_hand) AS inventory_value
FROM PRODUCT p
INNER JOIN INVENTORY i ON p.inventory_id = i.inventory_id
INNER JOIN CATEGORY c ON p.category_id = c.category_id
LEFT JOIN SUPPLIER s ON EXISTS (
    SELECT 1 FROM HARDWARE_PRODUCT hp
    INNER JOIN SUPPLIER_HARDWARE sh ON hp.h_id = sh.h_id
    WHERE hp.product_id = p.product_id AND sh.supplier_id = s.supplier_id
);

```

```

]-- VIEW 3: Employee Performance Summary
-- This view shows employee performance metrics
CREATE VIEW vw_EmployeePerformance
AS
SELECT
    e.employee_id,
    e.first_name + ' ' + e.last_name AS employee_name,
    e.position,
    e.hire_date,
    e.salary,
    COUNT(DISTINCT so.order_id) AS orders_processed,
    ISNULL(SUM(so.total_amount), 0) AS total_sales_handled,
    ISNULL(AVG(so.total_amount), 0) AS average_order_value,
    CASE
        WHEN COUNT(DISTINCT so.order_id) = 0 THEN 'No Sales'
        WHEN COUNT(DISTINCT so.order_id) >= 50 THEN 'High Performer'
        WHEN COUNT(DISTINCT so.order_id) >= 20 THEN 'Good Performer'
        ELSE 'Average Performer'
    END AS performance_rating
FROM EMPLOYEE e
LEFT JOIN SALES_ORDER so ON e.employee_id = so.employee_id
GROUP BY e.employee_id, e.first_name, e.last_name, e.position, e.hire_date, e.salary;

```

```

]-- VIEW 4: Supplier Analysis
-- This view provides supplier performance and relationship analysis
CREATE VIEW vw_SupplierAnalysis
AS
SELECT
    s.supplier_id,
    s.supplier_name,
    s.contact_person,
    s.email,
    COUNT(DISTINCT po.purchase_id) AS total_purchase_orders,
    ISNULL(SUM(po.total_amount), 0) AS total_purchase_value,
    ISNULL(AVG(po.total_amount), 0) AS average_order_value,
    COUNT(CASE WHEN po.status = 'Delivered' THEN 1 END) AS delivered_orders,
    COUNT(CASE WHEN po.status = 'Cancelled' THEN 1 END) AS cancelled_orders,
    CASE
        WHEN COUNT(DISTINCT po.purchase_id) = 0 THEN 0
        ELSE CAST(COUNT(CASE WHEN po.status = 'Delivered' THEN 1 END) * 100.0 / COUNT(DISTINCT po.purchase_id) AS DECIMAL(5,2))
    END AS delivery_success_rate
FROM SUPPLIER s
LEFT JOIN PURCHASE_ORDER po ON s.supplier_id = po.supplier_id
GROUP BY s.supplier_id, s.supplier_name, s.contact_person, s.email;

```

## Procedure Statements

```
-- PROCEDURE 1: Process New Sales Order
-- This procedure handles the complete process of creating a new sales order
CREATE PROCEDURE sp_ProcessNewSalesOrder
    @customer_id INT,
    @employee_id INT,
    @order_items NVARCHAR(MAX), -- JSON format: [{"product_id":1,"quantity":5,"unit_price":1500.00}]
    @payment_method NVARCHAR(20) = NULL,
    @payment_amount DECIMAL(10,2) = NULL,
    @order_id INT OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRANSACTION;

    BEGIN TRY
        DECLARE @total_amount DECIMAL(12,2) = 0;
        DECLARE @h_id INT = 1; -- Assuming hardware shop ID is 1

        -- Create the sales order
        INSERT INTO SALES_ORDER (order_date, total_amount, payment_status, order_type, employee_id, h_id)
        VALUES (GETDATE(), 0, 'Pending', 'Sale', @employee_id, @h_id);

        SET @order_id = SCOPE_IDENTITY();

        -- Parse and process order items (simplified - would need JSON parsing in real implementation)
        -- For demo purposes, we'll create a sample order item
        DECLARE @sample_quantity INT = 2;
        DECLARE @sample_price DECIMAL(10,2) = 1500.00;
        DECLARE @line_total DECIMAL(10,2) = @sample_quantity * @sample_price;

        -- Insert order item
        INSERT INTO ORDER_ITEM (quantity, unit_price, line_total, h_id)
        VALUES (@sample_quantity, @sample_price, @line_total, @h_id);

        DECLARE @order_item_id INT = SCOPE_IDENTITY();

        -- Link order and order item
        INSERT INTO SALES_ORDER_ORDER_ITEM (order_id, order_item_id)
        VALUES (@order_id, @order_item_id);

        SET @total_amount = @line_total;

        -- Update order total
        UPDATE SALES_ORDER
        SET total_amount = @total_amount
```

```

SET total_amount = @total_amount
WHERE order_id = @order_id;

-- Process payment if provided
IF @payment_method IS NOT NULL AND @payment_amount IS NOT NULL
BEGIN
    INSERT INTO PAYMENT (payment_date, payment_method, amount_paid, h_id, order_id)
    VALUES (GETDATE(), @payment_method, @payment_amount, @h_id, @order_id);

    DECLARE @payment_id INT = SCOPE_IDENTITY();

    -- Link customer credit
    INSERT INTO CUSTOMER_CREDIT (customer_id, order_id, payment_id, h_id, credit_limit)
    VALUES (@customer_id, @order_id, @payment_id, @h_id, 0);
END;

COMMIT TRANSACTION;
PRINT 'Sales order processed successfully. Order ID: ' + CAST(@order_id AS VARCHAR(10));

END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW;
END CATCH
END;
GO

```

```

-- PROCEDURE 2: Generate Inventory Reorder Report
-- This procedure generates a report of products that need to be reordered
CREATE PROCEDURE sp_GenerateReorderReport
AS
BEGIN
    SET NOCOUNT ON;

    SELECT
        p.product_id,
        p.product_name,
        p.brand,
        c.category_name,
        i.quantity_on_hand,
        i.minimum_stock_level,
        p.reorder_level,
        s.supplier_name,
        s.contact_person,
        sp.phone AS supplier_phone,
        (p.reorder_level - i.quantity_on_hand) AS suggested_order_quantity,
        (p.unit_price * (p.reorder_level - i.quantity_on_hand)) AS estimated_cost
    FROM PRODUCT p
    INNER JOIN INVENTORY i ON p.inventory_id = i.inventory_id
    INNER JOIN CATEGORY c ON p.category_id = c.category_id
    LEFT JOIN SUPPLIER_HARDWARE sh ON sh.h_id = 1 -- Hardware shop ID
    LEFT JOIN SUPPLIER s ON sh.supplier_id = s.supplier_id
    LEFT JOIN SUPPLIER_PHONE sp ON s.supplier_id = sp.supplier_id
    WHERE i.quantity_on_hand <= i.minimum_stock_level
    ORDER BY (i.minimum_stock_level - i.quantity_on_hand) DESC;

    -- Summary statistics
    SELECT
        COUNT(*) AS total_products_to_reorder,
        SUM(p.unit_price * (p.reorder_level - i.quantity_on_hand)) AS total_estimated_cost
    FROM PRODUCT p
    INNER JOIN INVENTORY i ON p.inventory_id = i.inventory_id
    WHERE i.quantity_on_hand <= i.minimum_stock_level;
END;
GO

```

```

-- PROCEDURE 3: Customer Credit Management
-- This procedure manages customer credit limits and outstanding balances
CREATE PROCEDURE sp_ManageCustomerCredit
    @customer_id INT,
    @action NVARCHAR(20), -- 'CHECK', 'UPDATE', 'INCREASE', 'DECREASE'
    @amount DECIMAL(10,2) = NULL,
    @current_balance DECIMAL(10,2) OUTPUT,
    @available_credit DECIMAL(10,2) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @total_credit_limit DECIMAL(10,2);
    DECLARE @total_outstanding DECIMAL(10,2);

    -- Calculate current credit limit and outstanding balance
    SELECT
        @total_credit_limit = ISNULL(SUM(cc.credit_limit), 0),
        @total_outstanding = ISNULL(SUM(so.total_amount - ISNULL(p.amount_paid, 0)), 0)
    FROM CUSTOMER_CREDIT cc
    LEFT JOIN SALES_ORDER so ON cc.order_id = so.order_id
    LEFT JOIN PAYMENT p ON cc.payment_id = p.payment_id
    WHERE cc.customer_id = @customer_id;

    SET @current_balance = @total_outstanding;
    SET @available_credit = @total_credit_limit - @total_outstanding;

    -- Perform requested action
    IF @action = 'CHECK'
    BEGIN
        -- Just return the calculated values
        SELECT
            @customer_id AS customer_id,
            @total_credit_limit AS total_credit_limit,
            @current_balance AS outstanding_balance,
            @available_credit AS available_credit;
    END
    ELSE IF @action = 'UPDATE' AND @amount IS NOT NULL
    BEGIN
        -- Update credit limit for all records of this customer
        UPDATE CUSTOMER_CREDIT
        SET credit_limit = @amount
        WHERE customer_id = @customer_id;

        PRINT 'Credit limit updated successfully';
    END
END

```

```

ELSE IF @action = 'INCREASE' AND @amount IS NOT NULL
BEGIN
    -- Increase credit limit
    UPDATE CUSTOMER_CREDIT
    SET credit_limit = credit_limit + @amount
    WHERE customer_id = @customer_id;

    PRINT 'Credit limit increased by ' + CAST(@amount AS VARCHAR(20));
END
ELSE IF @action = 'DECREASE' AND @amount IS NOT NULL
BEGIN
    -- Decrease credit limit (ensure it doesn't go below outstanding balance)
    UPDATE CUSTOMER_CREDIT
    SET credit_limit = CASE
        WHEN (credit_limit - @amount) < @current_balance
        THEN @current_balance
        ELSE (credit_limit - @amount)
    END
    WHERE customer_id = @customer_id;

    PRINT 'Credit limit decreased by ' + CAST(@amount AS VARCHAR(20));
END;
END;
GO

```

```

<-- PROCEDURE 4: Generate Business Intelligence Report
-- This procedure generates comprehensive business intelligence reports
CREATE PROCEDURE sp_GenerateBusinessReport
    @report_type NVARCHAR(20), -- 'SALES', 'INVENTORY', 'CUSTOMERS', 'SUPPLIERS'
    @start_date DATE = NULL,
    @end_date DATE = NULL
AS
BEGIN
    SET NOCOUNT ON;

    -- Set default date range if not provided
    IF @start_date IS NULL SET @start_date = DATEADD(month, -1, GETDATE());
    IF @end_date IS NULL SET @end_date = GETDATE();

    IF @report_type = 'SALES'
    BEGIN
        -- Sales Performance Report
        SELECT
            'Sales Summary' AS report_section,
            COUNT(DISTINCT so.order_id) AS total_orders,
            SUM(so.total_amount) AS total_revenue,
            AVG(so.total_amount) AS average_order_value,
            COUNT(DISTINCT cc.customer_id) AS unique_customers
        FROM SALES_ORDER so
        LEFT JOIN CUSTOMER_CREDIT cc ON so.order_id = cc.order_id
        WHERE so.order_date BETWEEN @start_date AND @end_date;

        -- Top performing employees
        SELECT
            'Top Employees' AS report_section,
            e.employee_id,
            e.first_name + ' ' + e.last_name AS employee_name,
            COUNT(so.order_id) AS orders_processed,
            SUM(so.total_amount) AS total_sales
        FROM EMPLOYEE e
        LEFT JOIN SALES_ORDER so ON e.employee_id = so.employee_id
        WHERE so.order_date BETWEEN @start_date AND @end_date
        GROUP BY e.employee_id, e.first_name, e.last_name
        ORDER BY total_sales DESC;
    END
    ELSE IF @report_type = 'INVENTORY'
    BEGIN
        -- Inventory Analysis Report
        SELECT * FROM vw_InventoryDashboard
        WHERE stock_status IN ('Out of Stock', 'Low Stock')
        ORDER BY
            CASE stock_status

```

```
CASE stock_status
    WHEN 'Out of Stock' THEN 1
    WHEN 'Low Stock' THEN 2
    ELSE 3
END;
ELSE IF @report_type = 'CUSTOMERS'
BEGIN
    -- Customer Analysis Report
    SELECT * FROM vw_CustomerSalesSummary
    ORDER BY total_purchases DESC;
END
ELSE IF @report_type = 'SUPPLIERS'
BEGIN
    -- Supplier Performance Report
    SELECT * FROM vw_SupplierAnalysis
    ORDER BY total_purchase_value DESC;
END;
GO
```

## Section 4

### Hardware Management System Application

#### Application Drive Link

[https://drive.google.com/drive/folders/19xoONBMd16\\_ZmSIhr-XVZdUcM9os3Tvl?usp=sharing](https://drive.google.com/drive/folders/19xoONBMd16_ZmSIhr-XVZdUcM9os3Tvl?usp=sharing)

#### Application Features

##### ◆ Filters

- Month selector for viewing sales by month
- Customer dropdown to filter data

##### ◆ Monthly Sales Summary

- Shows total sales, orders & revenue

##### ◆ Top Employees (Bar Chart)

- Ranks employees by sales

##### ◆ Sales Trends (Line/Bar Chart)

- Visualizes monthly sales data

##### ◆ Stock vs Sales (Pie Chart)

- Compares sold vs remaining inventory

##### ◆ Inventory Table

- Editable list with stock & sales info

##### ◆ Customer Section

- Edit customer details & view sales summary

##### ◆ Product Section

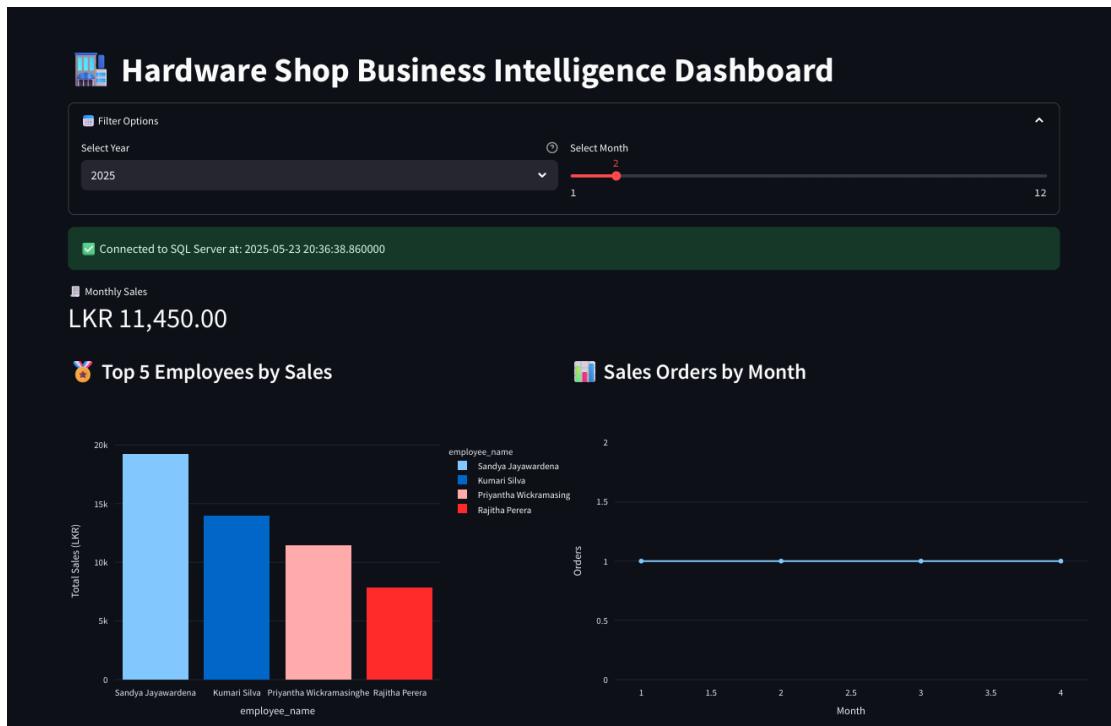
- Manage product name, price & stock

## Technologies Used

Layer	Technology	Purpose
Frontend	Streamlit	Web UI framework for data apps
Database	Microsoft SQL Server	Backend relational database
Database API	SQLAlchemy + pyodbc	ORM and ODBC driver to connect Python to SQL Server
Visualization	Plotly and Matplotlib	Interactive and static charts
Data Handling	Pandas	Data transformation and tabular display
Authentication	Windows Authentication	Used to access SQL Server without storing passwords

# User Interface Design

## Dashboard



### All Customers

	customer_id	first_name	last_name	email	address	customer_type	registration_date
0	1	Amara	Silva	amara.silva@email.com	12 Temple Road, Colombo 03	Individual	2024-01-15
1	2	Kasun	Perera	kasun.perera@email.com	45 Lake Drive, Kandy	Contractor	2024-01-20
2	3	Nimal	Fernando	nimal.fernando@email.com	78 Beach Road, Negombo	Business	2024-02-05
3	4	Sunitha	Jayawardena	sunitha.j@email.com	23 Hill Street, Nuwara Eliya	Individual	2024-02-10
4	5	Ruwan	Wickramasinghe	ruwan.w@email.com	67 Main Road, Galle	Contractor	2024-02-15
5	6	Chamara	Rajapaksa	chamara.r@email.com	89 Station Road, Matara	Individual	2024-03-01
6	7	Dilini	Gunawardena	dilini.g@email.com	34 Park Lane, Colombo 07	Business	2024-03-05
7	8	Pradeep	Amarasinghe	pradeep.a@email.com	56 School Lane, Kurunegala	Individual	2024-03-10
8	9	Malini	Dissanayake	malini.d@email.com	78 Church Street, Badulla	Contractor	2024-03-15
9	10	Saman	Rathnayake	saman.r@email.com	90 Market Street, Anuradhapura	Individual	2024-03-20

Select Customer

1 - Amara Silva

Edit Customer

### All Products

	product_id	product_name	brand	unit_price	reorder_level	unit_of_measure	description	order_item_id	inventory_id	category_id
0	1	Claw Hammer 16oz	Stanley	1500	20	PCS	Professional claw hammer with fiberglass handle	1	1	1
1	2	Cordless Drill 18V	Bosch	3500	10	PCS	Lithium-ion cordless drill with 2 batteries	2	2	2
2	3	Portland Cement 50kg	Holcim	250	100	BAG	High quality portland cement for construction	3	3	3
3	4	Circuit Breaker 20A	Siemens	1200	25	PCS	Single pole circuit breaker 20 ampere	4	4	4
4	5	PVC Pipe 4inch	Astral	450	50	MTR	High quality PVC pipe for plumbing	5	5	5
5	6	Angle Grinder 4.5inch	Makita	8500	5	PCS	Professional angle grinder with safety guard	6	6	2
6	7	Wood Screws 2inch	Generic	180	200	BOX	Phillips head wood screws pack of 100	7	7	6
7	8	Safety Helmet	MSA	720	30	PCS	Industrial safety helmet with chin strap	8	8	9
8	9	Wall Paint White 4L	Dulux	950	40	CAN	Premium quality interior wall paint	9	9	7
9	10	Garden Spade	Fiskars	650	15	PCS	Steel blade garden spade with wooden handle	10	10	8

Select Product

1 - Claw Hammer 16oz

Edit Product

### Customer Sales Summary

	customer_id	customer_name	email	customer_type	total_orders	total_purchases	average_order_value	last_order_date	customer_status
0	1	Amara Silva	amara.silva@email.com	Individual	0	0	0	None	Inactive
1	2	Kasun Perera	kasun.perera@email.com	Contractor	4	52620	13155	2025-04-08	Active
2	3	Nimal Fernando	nimal.fernando@email.com	Business	0	0	0	None	Inactive
3	4	Sunitha Jayawardena	sunitha.j@email.com	Individual	0	0	0	None	Inactive
4	5	Ruwan Wickramasinghe	ruwan.w@email.com	Contractor	2	30170	15085	2024-10-31	Occasional
5	6	Chamara Rajapaksa	chamara.r@email.com	Individual	0	0	0	None	Inactive
6	7	Dilini Gunawardena	dilini.g@email.com	Business	2	22640	11320	2024-12-14	Occasional
7	8	Pradeep Amarasinghe	pradeep.a@email.com	Individual	0	0	0	None	Inactive
8	9	Malini Dissanayake	malini.d@email.com	Contractor	2	30650	15325	2025-02-05	Occasional
9	10	Saman Rathnayake	saman.r@email.com	Individual	0	0	0	None	Inactive

## Customer Update

customer_id	first_name	last_name	email	address	customer_type	registration_date
2	3	Nimal Fernando	nimal.fernando@email.com	78 Beach Road, Negombo	Business	2024-02-05
3	4	Sunitha Jayawardena	sunitha.j@email.com	23 Hill Street, Nuwara Eliya	Individual	2024-02-10
4	5	Ruwan Wickramasinghe	ruwan.w@email.com	67 Main Road, Galle	Contractor	2024-02-15
5	6	Chamara Rajapaksa	chamara.r@email.com	89 Station Road, Matara	Individual	2024-03-01
6	7	Dilini Gunawardena	dilini.g@email.com	34 Park Lane, Colombo 07	Business	2024-03-05
7	8	Pradeep Amarasinghe	pradeep.a@email.com	56 School Lane, Kurunegala	Individual	2024-03-10
8	9	Malini Dissanayake	malini.d@email.com	78 Church Street, Badulla	Contractor	2024-03-15
9	10	Samani Rathnayake	saman.r@email.com	90 Market Street, Anuradhapura	Individual	2024-03-20
10	11	Thilani Mendis	thilani.m@email.com	12 River Road, Polonnaruwa	Business	2024-04-01
11	12	Ajith Bandara	ajith.b@email.com	45 Forest Lane, Hambantota	Contractor	2024-04-05

Select Customer

1 - Amara Silva

Edit Customer

First Name  
Amara

Last Name  
Silva

Email  
amara.silva@email.com

Address  
12 Temple Road, Colombo 03

Customer Type  
Individual

Update Customer

Customer updated.

## Product Delete

Brand  
Stanley

Unit Price  
1500.00

Reorder Level  
20

Unit of Measure  
PCS

Description  
Professional claw hammer with fiberglass handle

Category ID  
1

Inventory ID  
1

Update Product

Delete Product

Product deleted.

## Conclusion

The implementation of a hardware management system using MySQL (PL/SQL) and Python provides an efficient and reliable solution for managing the day-to-day operations of a retail hardware business. By organizing data across well-structured entities such as customers, employees, products, suppliers, sales orders, and payments, the system ensures accuracy, consistency, and easy retrieval of critical business information.

This system not only streamlines core processes such as inventory tracking, sales recording, and supplier management but also supports decision-making through real-time reporting and analytics. With the ability to handle high-volume transactions, multiple payment methods, and credit sales, it caters effectively to both individual customers and contractors.

Using Python for backend logic adds flexibility and automation, while MySQL's relational database structure guarantees data integrity. Overall, the proposed solution supports operational efficiency, reduces manual errors, and contributes to the long-term growth and professionalism of Jalitha Hardware.

## Contribution Table

Student ID	Name	Contribution
28666	W L C Wickramasinghe	10%
28466	R P J D Rajapaksha	10%
28510	K L D P P Kakulandala	10%
28663	M A J C B Dissanayake	10%
28131	H A J N Heshani	10%
28230	W H M N Jayawardhana	10%
28548	K S T Kavindi	10%
27595	H K I Dhananjaya	10%
28372	R V I Rathnayake	10%
28340	U V N S Mayuranga	10%