NORTHERN ILLINOIS UNIVERSITY

DE KALB, ILLINOIS

MAY 2025

Warehouse Operations Optimization through an In-House Management System

BY

MD ABU NAEEM KHAN

©2025 Md Abu Naeem Khan

A REPORT SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE EIS

MASTER OF SCIENCE

DEPARTMENT OF INDUSTRIAL AND

SYSTEMS ENGINEERING

Project Director:

Dr. Niechen Chen

**Executive Summary**

An once paper‑clogged warehouse now runs on a lean, Access‑powered database. The project began by mapping the daily headaches—missing inventory, slow picks, and guess‑work staffing. From there, a clean system was built, normalized to 4 NF, and ready for real‑time work. Every new shipment is logged the moment it hits the dock; stock levels update on the spot; each customer order instantly becomes a pick list assigned to one employee. With one click, managers can see what’s low, what’s waiting, and what’s selling fastest. Manual errors have plummeted, fulfillment moves quicker, and this small warehouse finally has room to grow.

Contents

[1. Introduction 3](#_Toc197283063)

[1.1 Problem Description 3](#_Toc197283064)

[1.2 Project Objectives and Scope 3](#_Toc197283071)

[2. Needs Analysis 4](#_Toc197283072)

[2.1 Needs and Requirements 5](#_Toc197283073)

[2. Real-time access to inventory availability to reduce stockouts and overstocking. 5](#_Toc197283075)

[3. Automated generation of pick lists based on active orders to reduce human error. 5](#_Toc197283076)

[4. Ability to assign and track picker staff against each order for better workload management. 5](#_Toc197283077)

[2.2 Gap Analysis: 5](#_Toc197283078)

[3.1 Process Model for Current State 6](#_Toc197283079)

[3.2 Current ER Model 8](#_Toc197283080)

[3.3 Obligatory/Non-Obligatory Analysis 9](#_Toc197283081)

[3.4 Data Normalization 10](#_Toc197283082)

[4. Answering Business Questions by SQL Querying 12](#_Toc197283083)

[5. Conclusion 13](#_Toc197283084)

## 1. Introduction

### 1.1 Problem Description

### The warehouse under study operates with a fully manual system, relying on paper-based logs and spreadsheet files to manage its inventory, order processing, and staff assignments. This outdated system creates a variety of operational challenges that directly impact efficiency, accuracy, and customer satisfaction.

### One of the most critical issues is the absence of real-time inventory visibility. Since stock levels are updated manually and tracked through spreadsheets, there are frequent inventory mismatches between recorded and actual stock. This often leads to stockouts, overstocking, and incorrect shipments. As a result, orders are sometimes delayed or fulfilled inaccurately, creating poor customer experience and increasing the risk of revenue loss. Order processing is also heavily affected by the lack of automation. Staff members prepare pick lists manually, sometimes using outdated or incomplete information. This results in errors during the picking process, such as missing or incorrect items being included in outbound orders. Moreover, when discrepancies are found, warehouse employees must repeat tasks, which consume time and resources and causes additional bottlenecks in workflow.

### Staff workload tracking and task assignment are performed informally or not at all. Without a centralized system, it is difficult for the warehouse manager to know which employee handled which order or how many pick lists are assigned to each staff member. This leads to imbalanced workloads, inefficiencies in task distribution, and poor accountability.

### The manual nature of the system also creates challenges in maintaining data integrity. Errors such as duplicate records, mis keyed entries, and missing data are common and difficult to detect until they impact operations. There is no enforcement of data relationships or consistency across records, which undermines the reliability of reports and decision-making.

### Furthermore, operational reporting is either non-existent or compiled manually from fragmented sources. This delays the ability of the warehouse manager to identify low-stock items, pending orders, and staff performance. Decision-making is therefore reactive rather than proactive

### Collectively, these issues demonstrate a critical need for a computerized warehouse management system that can automate data flows, enforce consistency, and provide real-time visibility into warehouse operations. Without such a system, the warehouse cannot scale, maintain service quality, or make efficient use of its human and physical resources.

### 1.2 Project Objectives and Scope

The main objective of this project is to optimize the warehouse operations by replacing the existing manual system with an automated in-house information system. Specially, the project aim to:

1. Automating key warehouse tasks like inventory keeping and order fulfillment will cut down on mistakes made by hand.
2. Allowing managers and employees to view stock levels in real time so they are always aware of what is available and what is running short.
3. Creating automated pick lists and clearly defining tasks to assist warehouse employees in completing orders more quickly.
4. Consolidating all warehouse data into a singular centralized MS Access database, rather than utilizing disparate spreadsheets.
5. Building relevant queries and reports to address operational questions like which items are low or who picked the most orders.

The scope of this project was intentionally kept small in order to concentrate on the warehouse's internal operations. Among the primary functional areas were:

1. Monitoring product stock levels and keeping track of incoming shipments (Inventory management).
2. Order fulfillment is the process of handling product movement from stock and processing customer orders.
3. Pick jobs are delegated to warehouse employees, and work distribution is tracked (Staff assignment).
4. Creating queries in MS Access to monitor operational status and performance indicators.

External company operations like financial transactions, supplier procurement, and customer relationship management were not included in the system.

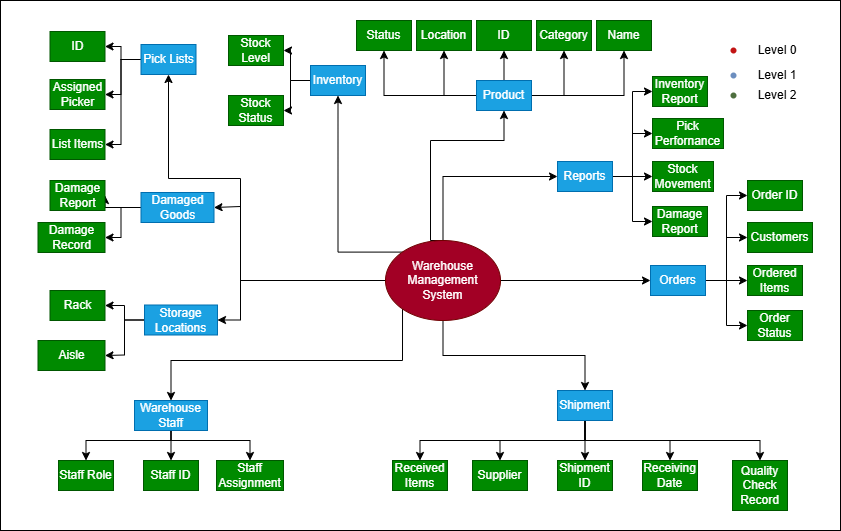


Figure 1: Concept Map

## 2. Needs Analysis

Because of the old, manual methods that were in place, the warehouse manager had to deal with a number of practical issues. Handwritten logs and spreadsheets were used for everything, from keeping track of supplies to picking orders. This made the system error-prone, inconsistent, and time-consuming. It was hard to answer even simple business questions like "What's low in stock?" or "Which orders are still pending?" because there wasn't a central place to see how the warehouse was running as a whole.

A concept map was made to show all the important parts of warehouse operations so that these issues could be better understood and solved. This included things like products, inventory levels, staff jobs, pick lists, shipment records, and how they all had to work together with the new system. The concept map shows that the warehouse management system should be the hub that all the important processes and data flows connect to. The system would keep track of new products and stock, make pick lists, assign work to staff, and give organized reports. It became clearer what the system needed to do and what was missing from the manual setup after the information was laid out clearly.

### 2.1 Needs and Requirements

Previously, the warehouse was run entirely by hand, using spreadsheets and handwritten diaries to track personnel actions, inventory levels, and product information. This approach frequently resulted in inconsistent or missing data, delayed order fulfillment, and stock problems. In order to increase precision, effectiveness, and visibility, the following demands and specifications were determined:

The requirements are as follows:

### A centralized system where all product, inventory, order, and staff data are stored and managed.

### Real-time access to inventory availability to reduce stockouts and overstocking.

### Automated generation of pick lists based on active orders to reduce human error.

### Ability to assign and track picker staff against each order for better workload management.

### 2.2 Gap Analysis:

After the needs and requirements, a gap analysis of the main needs identified, specifying the current standard (As Is) and expected standard (To Be), and the gap between both.

|  |  |  |
| --- | --- | --- |
| **As Is (Manual System)** | **To Be (Proposed System)** | **Gap** |
| Inventory updated manually in spreadsheets | Inventory managed and updated in real-time through Access tables | Real-time stock visibility |
| Orders and picking managed using paper-based logs | Pick lists auto-generated and linked to staff and orders in the system | Reduced errors in picking and delays |
| No structured way to track staff workload | Picker assignments linked and tracked via database relationships | Improved accountability and task tracking |

Realizing that this would all count towards the final proposal that he would present to the management regarding information system management, the IE made a detailed process model with a structure notation and identified the time taken for each process. He figured this would help in arguing for a unified information system later in the process.

**3. System Design**

### 3.1 Process Model for Current State

Following the identification of basic requirements and the execution of a gap analysis, the subsequent phase of the project concentrated on the comprehensive design of the warehouse management system utilizing Data Flow Diagrams (DFDs). This step entailed mapping the data flow across diverse warehouse tasks, allowing the Industrial Engineer to view existing operations and find opportunities for automation and integration to enhance efficiency.

DFD modeling adhered to a hierarchical framework, wherein processes were systematically divided into increasingly specific subprocesses. The diagrams provide a basis for database design, form creation, and query logic in MS Access.

Level 0 Context-Level Process Diagram

At the context level, the system is visualized as a single process: “Warehouse Management System.” This diagram captures the primary external entities-Shipments (from suppliers) and Orders (from customers)-that provide inputs to the system. The system processes these inputs to produce two key outputs:

* Fulfilled Orders (to be dispatched to customers)
* Updated Inventory (reflecting received and issued stock)

A diagram of a warehouse management system

AI-generated content may be incorrect.

Figure 2: Warehouse Data Analysis Level 0 Diagram

Level 1 Process Diagram  
The Level 0 process is decomposed into two main functional processes:

1.1 Receive Stock and Manage Inventory: Handles inbound shipments from suppliers. The process includes checking incoming goods, updating inventory levels, and recording the transaction.

1.2 Process Orders and Reporting: Manages customer orders by matching requests against stock, creating pick lists, and recording order fulfillment and inventory movements.

Each process updates the system records accordingly, ensuring real-time traceability of inventory status and customer fulfillment progress.

A diagram of a process

AI-generated content may be incorrect.

Figure 3: Warehouse Data Analysis Level 1 Diagram

Level 2(a) Process Diagram  
This level dives deeper into the 1.1 process. It includes:

1.1.1 Inspect and Receive Goods: Warehouse staff inspect shipments for accuracy, quality, and condition. Verified goods are accepted for storage.

1.1.2 Update Stock and Mark Exceptions: Inventory is adjusted based on received items. Damaged, returned, or mismatched items are flagged, and periodic cycle counts help reconcile actual stock with system records.

A diagram of a product

AI-generated content may be incorrect.

Figure 4: Warehouse Data Analysis Level 2 Diagram 1

Level 2(b) Process Diagram

This diagram details the 1.2 process:

1.2.1 Create Pick Lists: Once customer orders are received, the system generates pick lists that guide warehouse staff to retrieve items from storage efficiently.

1.2.2 Dispatch and Reporting: Items are packed and dispatched. Inventory is adjusted accordingly, and reports are generated to reflect fulfillment status and inventory impact.

.

A diagram of a customer order

AI-generated content may be incorrect.

Figure 5: Warehouse Data Analysis Level 2 Diagram 2

### 3.2 Current ER Model

Below is the ER model for expedited data analysis for Warehouse optimization.

A diagram of a company

AI-generated content may be incorrect.

Figure 6: Current ER Model

### 3.3 Obligatory/Non-Obligatory Analysis

In this section the obligatory/ non-obligatory analysis for the entities in ER model is detailed.

| **Step** | **Relationship** | **A side** | **B side** | **Memberships we observed** | **Table decision produced** |
| --- | --- | --- | --- | --- | --- |
| ① | **supply**(Supplier ↔ Product) | Product may exist before vendor chosen | Supplier may exist before carrying SKUs | optional / optional (**0 : 0‑N**) | Needs **three tables** but offers **no integrity value** → *Supplier* removed altogether to keep schema minimal. |
| ② | **stored**(Product ↔ Inventory) | Product can have zero stock | Every inventory row must map to a product | optional / obligatory (**1 : 0‑N**) | Keep **Product** & **Inventory**. Inventory.ProductID set **NOT NULL**. |
| ③ | **OrderItem** junction | Order **must** list ≥ 1 items | Product may never be ordered | obligatory / optional | Keep **OrderItem** with composite PK (OrderID, ProductID); both FKs **NOT NULL**. |
| ④ | **generates**(Order ↔ PickList) | Some orders are on hold, so no pick list yet | Every pick list belongs to exactly one order | optional / obligatory (**1 : 0‑1**) | Two separate tables; PickList.OrderID **NOT NULL**. |
| ⑤ | **pick**(WarehouseStaff ↔ PickList) | Staff may be idle | Each pick list needs a staff member | optional / obligatory (**1 : 0‑N**) | PickList.PickerID **NOT NULL** as FK to **WarehouseStaff**. |

Based on obligatory and non-obligatory analysis, supplier is removed. After the deleting the supplier, premilary database design are:

* Product (ProductID, ProductName, CategoryStatus)
* Order (OrderID, CustomerID, Order\_Date, Order\_Status)
* OrderItem (OrderID,ProductID, Quatity Ordered)
* Inventory (InventoryID, ProductID, QuantityAvailable)
* PickList (Picklist, OrderID, PickerID, Pick\_start\_time, Pick\_end\_time)
* WarehouseStaff (StaffID, StaffName, Role)

### 3.4 Data Normalization

Normalization ensures data is logically stored with minimal redundancy, improved integrity, and efficient access. Below is an explanation of how your database structure satisfies the first three normal forms (1NF, 2NF, 3NF, and 4NF), which is typically sufficient for most warehouse management systems.

| **Entity** | **Primary Key** | **1 NF(Atomic Attributes)** | **2 NF(No Partial‑Key Dependency)** | **3 NF(No Transitive Dependency)** | **4 NF(No Multi‑Valued Dependency)** |
| --- | --- | --- | --- | --- | --- |
| **Product** | ProductID | ProductID, ProductName, CategoryStatus are all single‑valued. | Non‑key attrs depend on ProductID. | No non‑key depends on another non‑key. | A product has one category/status →no multi‑valued facts. |
| **Inventory** | InventoryID | InventoryID, ProductID, QuantityAvailable are atomic. | QuantityAvailable depends only on InventoryID. | No transitive dependency. | Each inventory row links to one product only. |
| **Order** | OrderID | OrderID, CustomerID, OrderDate, OrderStatus are atomic. | All non‑key attrs depend on OrderID. | CustomerID does not determine other non‑keys. | Each order ties to one customer →no independent multi‑sets. |
| **OrderItem** | OrderID + ProductID | Composite row holds one ProductID and QuantityOrdered. | QuantityOrdered depends on the *full* composite key. | No transitive dependency among non‑keys. | Row records one product per order line → no multi‑valued facts. |
| **PickList** | PickListID | PickListID, OrderID, PickerID, PickStartTime, PickEndTime are atomic. | All non‑key attrs depend on PickListID. | No non‑key depends on another non‑key. | Each pick list linked to one order and one picker only. |
| **WarehouseStaff** | StaffID | StaffID, StaffName, Role are single‑valued. | Non‑keys depend solely on StaffID. | No transitive dependency. | One role per staff member → no multi‑valued dependency. |
| **Customer** | CustomerID | CustomerID, CustomerName, Address, Contact are atomic. | All non‑keys depend on CustomerID. | No transitive dependency. | One address/contact per customer in current scope. |

Final ER Model After obligatory, non obligatory analysis and normalization:

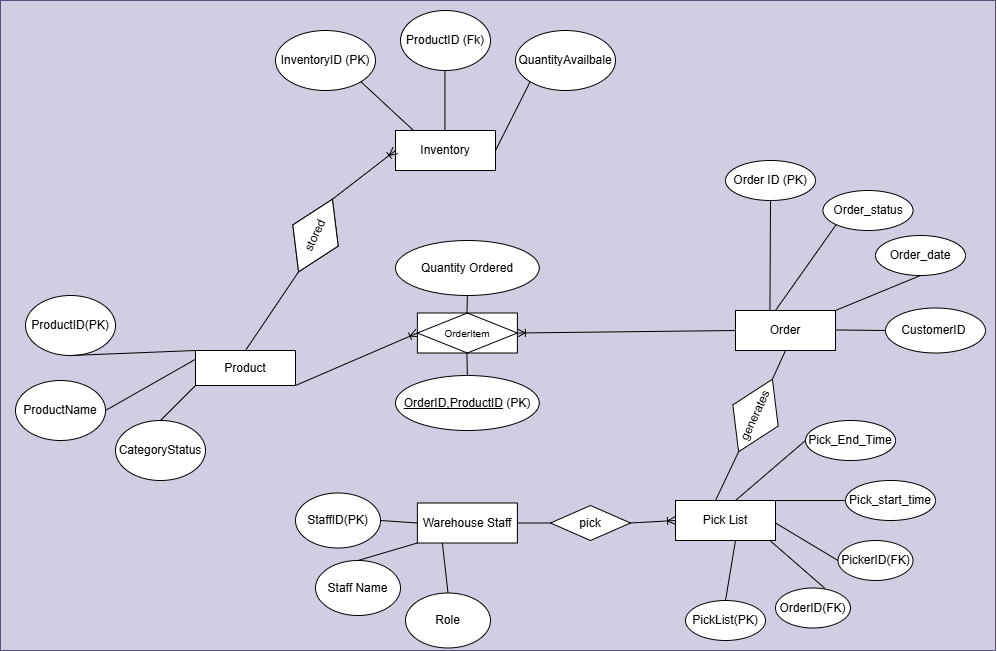


Figure 7: Normalized ER Model

And the same tables are created in MS Access below.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 8: Access ER Diagram

# 4. Answering Business Questions by SQL Querying

Which products currently have low inventory (less than 50 units)?A screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Which customer orders are still pending and need to be fulfilled?A screenshot of a computer

AI-generated content may be incorrect.

Which customers placed orders for which products, and how much quantity did they order?A screenshot of a computer

AI-generated content may be incorrect.

What are the top 5 products ordering the most (highest quantity ordered)?A screenshot of a computer

AI-generated content may be incorrect.

# 5. Conclusion

This project transformed a paper‑based, error‑prone warehouse into a fully digital operation supported by an in‑house Microsoft Access database. We mapped existing processes with multi‑level DFDs, designed and normalized a relational data model to 4 NF, and built tables, forms, queries, and reports that now provide real‑time inventory visibility, automated pick‑list generation, and reliable staff‑workload tracking. By centralizing product, order, and stock information, the system eliminates duplicate records, reduces manual entry mistakes, and cuts order‑fulfillment time. The project demonstrated the complete systems‑analysis life cycle: needs assessment, gap analysis, conceptual modeling, logical database design, and physical implementation. It also reinforced key concepts in data normalization, referential integrity, and SQL‑based decision support.  
  
Options for future improvements include barcode scanning for speedier data entry, remote access via web or mobile, and integration with supplier or customer portals for automated replenishment and order confirmation. Adding predictive inventory analytics or real-time dashboards to the reporting module enhances managerial insight. The project optimised warehouse operations and established a scalable platform for development and refinement.