

**UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI**

**INFORMATION COMMUNICATION TECHNOLOGY DEPARTMENT**

DATA SCIENCE – CLASS DS – GROUP 5

**Fundamental of database**

Final Report

**Warehouse**

**Lê Xuân Lộc - 22BI13256** [loclx.22bi13256@usth.edu.vn]

**Lê Duy Anh - 22BI13017** [anhld.22BI13017@usth.edu.vn]

**Nguyễn Văn Minh - 22BI13306** [minhnv.22bi13306@usth.edu.vn]

**Nguyễn Trần Minh Quân - 22BI13375** [quanntm.22bi13375@usth.edu.vn] **Lê Thuận Ninh - 22BI13354** [ninhlt.22bi13354@usth.edu.vn]

Hanoi, Vietnam, December 2023

**TABLE OF CONTENTS**

[INTRODUCTION 2](#_Toc152793761)

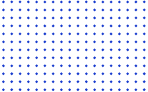
[ERD 4](#_Toc152793762)

[DATA SCHEMA 6](#_Toc152793763)

[STATEMENT 12](#_Toc152793764)

[Contribution and evaluation 16](#_Toc152793765)

# INTRODUCTION



The Warehouse Management System (WMS) plays a critical role in the efficient management and tracking of warehouse operations for enterprises. In today's fast-paced business environment, enterprises face increasing pressure to streamline their supply chain processes, optimize inventory management, and meet customer demands promptly. To address these challenges, an effective and comprehensive WMS is essential. This report provides a detailed overview of a Warehouse Management System database that is specifically designed to cater to the needs of enterprises and empower them to handle complex warehouse operations effectively.

Reasons for Enterprises to Adopt the Warehouse Management System Database:

1. Streamlined Inventory Management:

Inventory is a significant asset for any enterprise, and effective management is crucial for ensuring smooth operations and customer satisfaction. The Warehouse Management System database provides enterprises with a centralized platform to store and track inventory data. With real-time visibility into stock levels, item details, and movement history, enterprises can optimize inventory replenishment, reduce stockouts, minimize holding costs, and improve overall inventory accuracy.

2. Enhanced Operational Efficiency:

Efficient warehouse operations are vital for enterprises to meet customer expectations and deliver products in a timely and cost-effective manner. The Warehouse Management System database enables enterprises to streamline various processes, such as order fulfillment, picking, packing, and shipping. By automating and optimizing these operations, enterprises can improve order accuracy, minimize processing time, reduce labor costs, and enhance overall operational efficiency.

3. Improved Logistics and Shipment Processes:

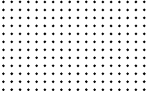
Effective logistics and shipment management are crucial for enterprises to ensure on-time delivery, minimize shipping errors, and reduce transportation costs. The Warehouse Management System database provides robust features to manage and track shipments, including sender and receiver details, item information, and shipping schedules. With this information readily available, enterprises can efficiently plan and coordinate shipments, track delivery status, and implement proactive measures to address any potential issues.

4. Data-Driven Decision Making:

In today's data-centric business landscape, enterprises rely on accurate and timely information to make informed decisions. The Warehouse Management System database captures and analyzes vast amounts of data related to warehouse operations, employee performance, inventory levels, and customer demand. By leveraging advanced analytics and reporting capabilities, enterprises can gain valuable insights into operational trends, identify areas for improvement, optimize resource allocation, and make data-driven decisions to enhance overall warehouse performance.

To conclude, the Warehouse Management System database presented in this report offers enterprises a comprehensive solution to effectively manage and track their warehouse operations. With streamlined inventory management, enhanced operational efficiency, improved logistics and shipment processes, and data-driven decision-making capabilities, enterprises can gain a competitive edge in today's dynamic business landscape. By adopting this robust database, enterprises can optimize their supply chain processes, meet customer demands efficiently, and drive overall business growth.

# ERD



1. **Definition**

An Entity Relationship Diagram (ERD) is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

**2. Relationship**

A diagram of a flowchart

Description automatically generated

**Employee:**

This table stores information about each employee, including their ID, full name, age, address, and the ID of the warehouse they are associated with. Each employee is uniquely identified by their ID.

**Items:**

This table stores information about the items that are being sent or received. It includes details such as the item’s ID, name, weight, size, quantity, value, the address it’s being sent from, the address it’s being sent to, and any additional details. Each item is uniquely identified by its ID.

**Sender:**

This table stores information about the sender of each item. It includes the sender’s ID, full name, phone number, address, the time the item was sent, and the ID, name, and quantity of the item being sent. Each sender is uniquely identified by their ID.

**Receiver:**

This table stores information about the receiver of each item. It includes the receiver’s ID, full name, phone number, address, and the ID, name, and quantity of the item being received. Each receiver is uniquely identified by their ID.

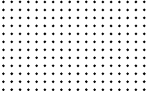
**Shipping:**

This table stores information about each shipping record. It includes the ID of the shipping record, the name of the driver, the space available in the truck, the order ID, and the ID of the warehouse. Each shipping record is uniquely identified by its ID.

**Warehouse:**

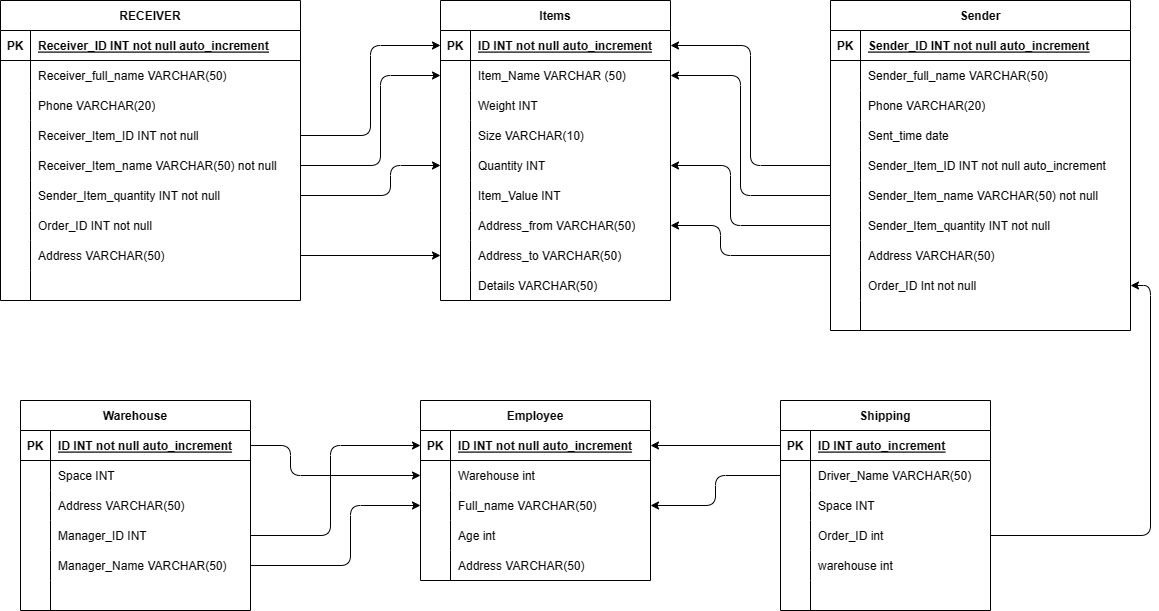
This table stores information about each warehouse, including its ID, available space, address, the name of the manager, and the manager’s ID. Each warehouse is uniquely identified by its ID. The relationships between these tables are represented by the foreign keys. For example, the Warehouse column in the Employee table is a foreign key that references the ID column in the Warehouse table. This means that each employee is associated with a warehouse.

# DATA SCHEMA



The database schema consists of six tables: Employee, Warehouse, Truck, Item, Sender, and Receiver. These tables define the entities, attributes, and relationships in the database, allowing for the storage and tracking of employees, warehouses, trucks, items, senders, and receivers. The tables are connected through foreign key relationships, which establish associations between related entities. The primary keys ensure the uniqueness of each record within a table.

By understanding the structure and relationships defined in the schema, you can efficiently store, retrieve, and manage data related to employees, warehouses, trucks, items, senders, and receivers in your database system.



**1. Table: Employee**

- Columns:

- ID (INT, auto\_increment, not null): Represents the unique identifier for an employee.

- Full\_name (VARCHAR(50)): Stores the full name of an employee.

- Age (INT): Represents the age of an employee.

- Address (VARCHAR(50)): Stores the address of an employee.

- Warehouse (VARCHAR(50)): Stores the name of the warehouse associated with the employee.

- Primary Key: ID

**2. Table: Items**

- Columns:

- ID (INT, auto\_increment, not null): Represents the unique identifier for an item.

- Item\_Name (VARCHAR(50)): Stores the name of the item.

- Weight (INT): Represents the weight of the item.

- Size (VARCHAR(10)): Represents the size of the item.

- Quantity (INT): Stores the quantity of the item.

- Item\_Value (INT): Represents the value of the item.

- Address\_from (VARCHAR(50)): Stores the address from where the item is being sent.

- Address\_to (VARCHAR(50)): Stores the address to where the item is being sent.

- Details (VARCHAR(50)): Stores additional details about the item.

- Primary Key: ID

**3. Table: Sender**

- Columns:

- Sender\_ID (INT, auto\_increment, not null): Represents the unique identifier for a sender.

- Sender\_full\_name (VARCHAR(50)): Stores the full name of the sender.

- Phone (VARCHAR(20)): Stores the phone number of the sender.

- Address (VARCHAR(50)): Stores the address of the sender.

- Sent\_time (DATE): Represents the time when the item was sent.

- Sender\_Item\_ID (INT, auto\_increment, not null): Represents the unique identifier for the item being sent.

- Sender\_Item\_name (VARCHAR(50), not null): Stores the name of the item being sent.

- Sender\_Item\_quantity (INT, not null): Stores the quantity of the item being sent.

- Order\_ID (INT, not null): Represents the ID of the order associated with the sender.

- Foreign Keys:

- Sender\_Item\_ID (references Items(ID)): Represents a foreign key relationship to the ID column in the Items table, indicating the item being sent.

- Sender\_Item\_name (references Items(Item\_Name)): Represents a foreign key relationship to the Item\_Name column in the Items table, indicating the item being sent.

- Sender\_Item\_quantity (references Items(Quantity)): Represents a foreign key relationship to the Quantity column in the Items table, indicating the quantity of the item being sent.

- Address (references Items(Address\_from)): Represents a foreign key relationship to the Address\_from column in the Items table, indicating the address from where the item is being sent.

- Primary Key: Sender\_ID

**4. Table: RECEIVER**

- Columns:

- Receiver\_ID (INT, auto\_increment, not null): Represents the unique identifier for a receiver.

- Receiver\_full\_name (VARCHAR(50)): Stores the full name of the receiver.

- Phone (VARCHAR(20)): Stores the phone number of the receiver.

- Address (VARCHAR(50)): Stores the address of the receiver.

- Receiver\_Item\_ID (INT, not null): Represents the ID of the item being received.

- Receiver\_Item\_name (VARCHAR(50), not null): Stores the name of the item being received.

- Receiver\_Item\_quantity (INT, not null): Stores the quantity of the item being received.

- Order\_ID (INT, not null): Represents the ID of the order associated with the receiver.

- Foreign Keys:

- Receiver\_Item\_ID (references Items(ID)): Represents a foreign key relationship to the ID column in the Items table, indicating the item being received.

- Receiver\_Item\_name (references Items(Item\_Name)): Represents a foreign key relationship to the Item\_Name column in the Items table, indicating the item being received.

- Receiver\_Item\_quantity (references Items(Quantity)): Represents a foreign key relationship to the Quantity column in the Items table, indicating the quantity of the item being received.

- Address (references Items(Address\_to)): Represents a foreign key relationship to the Address\_to column in the Items table, indicating the address to where the item is being received.

- Primary Key: Receiver\_ID

**5. Table: Shipping**

- Columns:

- ID (INT, auto\_increment): Represents the unique identifier for shipping.

- Driver\_Name (VARCHAR(50)): Stores the name of the driver.

- Space (INT): Stores the available space in the shipping.

- Order\_ID (INT): Represents the ID of the order associated with the shipping.

- warehouse (INT): Represents the ID of the warehouse associated with the shipping.

- Foreign Keys:

- Driver\_Name (references Employee(Full\_name)): Represents a foreign key relationship tothe Full\_name column in the Employee table, indicating the driver associated with the shipping.

- ID (references Employee(ID)): Represents a foreign key relationship to the ID column in the Employee table, indicating the ID of the employee associated with the shipping.

- Order\_ID (references Sender(Order\_ID)): Represents a foreign key relationship to the Order\_ID column in the Sender table, indicating the order associated with the shipping.

- Primary Key: ID

**6. Table: Warehouse**

- Columns:

- ID (INT, auto\_increment, not null): Represents the unique identifier for a warehouse.

- Space (INT): Represents the available space in the warehouse.

- Address (VARCHAR(50)): Stores the address of the warehouse.

- Manager\_Name (VARCHAR(50)): Stores the name of the warehouse manager.

- Manager\_ID (INT): Represents the ID of the warehouse manager.

- Foreign Keys:

- Manager\_Name (references Employee(Full\_name)): Represents a foreign key relationship to the Full\_name column in the Employee table, indicating the manager associated with the warehouse.

- Manager\_ID (references Employee(ID)): Represents a foreign key relationship to the ID column in the Employee table, indicating the ID of the manager associated with the warehouse.

- Primary Key: ID

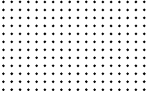
Additionally, an alteration is made to the Employee table:

ALTER TABLE Employee

ADD FOREIGN KEY (Warehouse) REFERENCES Warehouse(ID);

This ALTER TABLE statement adds a foreign key relationship from the Warehouse column in the Employee table to the ID column in the Warehouse table, indicating the warehouse associated with an employee.

# STATEMENT



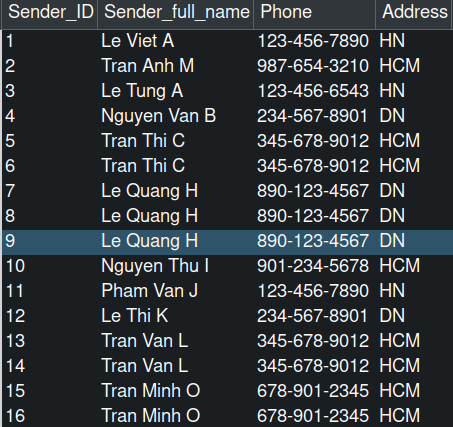
**Functionalities:**

Statement#1:

Select distinct Sender\_ID, Sender\_full\_name, Phone, Address from Sender;

-This statement display all the sender.

Output:



Statement#2:



-This statement allow us to delete any particular order.

Statement#3:

-This statement find all order that involve a specific item. In this case is ‘LyThuyTinh’.



Output:

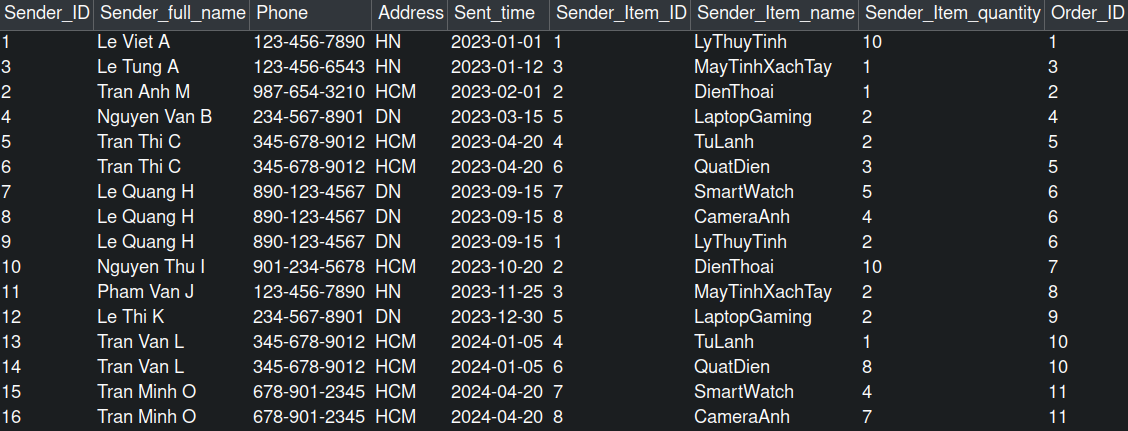


Statement#4:



-This statement show the queue of the priority of the orders base on time. The oldest order will be put on top and so on.

Output:

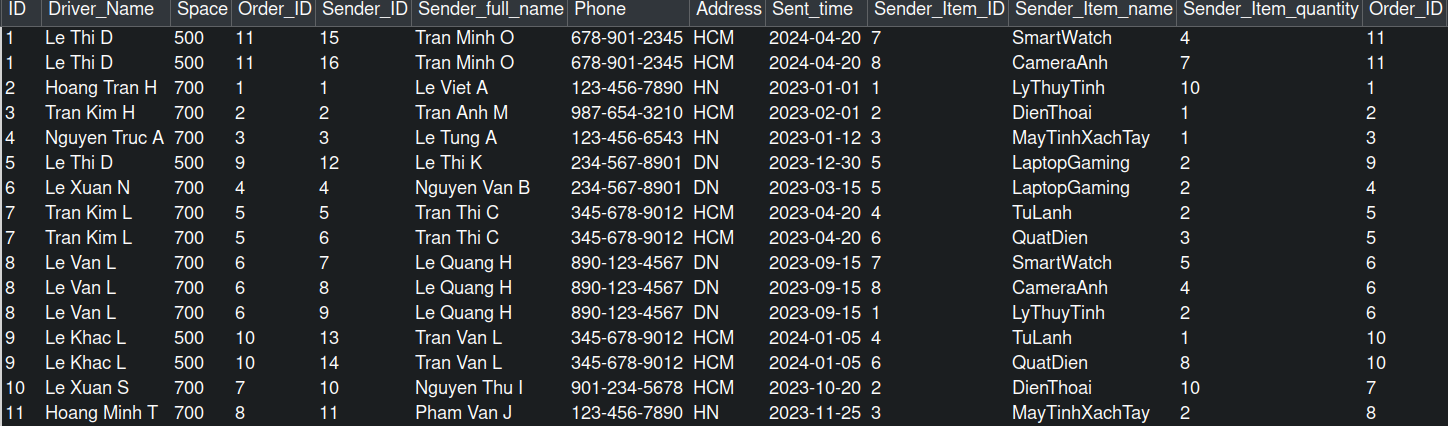


Statement#5:



-This statement helps find how many order each driver had to ship as well as the detail of that order.

Output:



Statement#6:

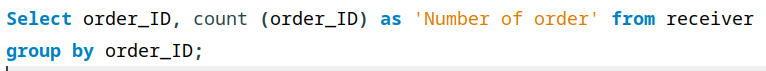


-This statement find out which employee is working at which warehouse. In this particular case, looking for information of employee working at warehouse 3.

Output:

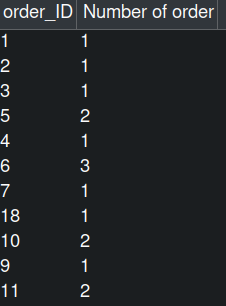


Statement#7:



-This statement helps find out the number of shipment needed for each order.

Output:



Statement#8:

-This statement show us the total amount of purchase.



Output:



# Contribution and evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| Full Name | Student ID | Contribution | Grades |
| **Lê Xuân Lộc - Leader** | 22BI13256 | 20% |  |
| **Lê Duy Anh** | 22BI13017 | 20% |  |
| **Nguyễn Trần Minh Quân** | 22BI13375 | 20% |  |
| **Nguyễn Văn Minh** | 22BI13306 | 20% |  |
| **Lê Thuận Ninh** | 22BI13354 | 20% |  |