



PROJECT PHASE 1

Course: Database Systems Lab

Section: A4

Submitted By

Hashir Hassan F2023332040

Muhammad Ahab Raza F2023332016



Estd. 1990



Contents

Introduction	1
Problem Statement	2
Project Scope	2
Tools and Technologies	2
Entities, Attributes, and Descriptions.....	3
1. Employee.....	3
2. Customer	3
3. Sale	3
4. Sale Item (<i>Weak Entity</i>)	3
5. Product.....	3
6. Category	3
7. Purchase Order	3
8. Purchase Item (<i>Weak Entity</i>)	3
9. Supplier	4
Entity Relationships and Cardinality	4
1. Employee — Records — Sale	4
2. Customer — Makes — Sale	4
3. Sale — Has — Sale Item	4
4. Sale Item — Has — Product	4
5. Product — Has — Category.....	4
6. Supplier — Gives — Purchase Order.....	5
7. Employee — Manages — Purchase Order	5
8. Purchase Order — Consists Of — Purchase Item.....	5
9. Purchase Item — Has — Product	5
Entity Relationship Diagram	6
Chen Model (Simple ERD):	6
Crow Foot Model (Relational Model):.....	7

Introduction

A Cash and Carry store is a type of retail business where customers buy products in bulk and make immediate payments. These stores often deal with a wide variety of items and a large number of transactions daily. Managing products, sales, suppliers, and inventory manually can lead to problems such as incorrect stock levels, missed restocking, and poor tracking of purchases and sales.

The purpose of this project is to develop a simple command-line-based management system that will help a Cash and Carry store keep better records of its products, suppliers, purchases, and sales. It will also help maintain inventory automatically, reduce human error, and save time.

Problem Statement

In many small to medium-sized retail businesses, especially cash and carry stores, day-to-day operations like managing inventory, recording sales, and handling purchases from suppliers are still done manually or using basic tools like notebooks or spreadsheets. This manual approach often leads to problems such as stock mismanagement, data duplication, missing sales records, and errors in tracking purchases or available inventory.

As the number of products and transactions grows, it becomes increasingly difficult to monitor product availability, supplier deliveries, and sales trends without a proper system in place. Store staff may forget to reorder out-of-stock items, or mistakenly sell items that are no longer in stock, leading to customer dissatisfaction and loss of revenue.

To solve this issue, there is a need for a simple yet effective management system that can automate essential processes like tracking sales, recording purchases, managing product data, updating stock levels, and keeping supplier and employee records. A command-line-based system connected to a database can help streamline operations, reduce human error, and allow store owners to maintain better control over their business.

Project Scope

This project focuses on designing and developing a basic management system for a Cash and Carry store using a command-line interface and a relational database. The system will handle core store operations such as managing product information, recording sales made to customers, tracking purchase orders from suppliers, and automatically updating inventory based on sales and purchases.

The system will allow employees to add, update, and delete product details, assign products to categories, and maintain accurate stock levels. It will also record each sale and purchase with the involved products and quantities, helping the store maintain a proper flow of inventory.

The scope of this project includes:

- Maintaining records of products, categories, suppliers, customers, and employees
- Recording customer sales and generating sale entries with multiple products
- Creating and managing purchase orders from suppliers
- Automatically adjusting inventory based on sales and purchases
- Designing an ERD (Chen and Crow's Foot models) to represent the system structure
- Implementing the system with Python (CLI) and MySQL

Advanced features such as barcode scanning, graphical user interface, online customer orders, and detailed financial reporting are not part of this phase and may be considered in future improvements.

Tools and Technologies

- **Python:** Used to build the command-line interface (CLI) for interacting with the system. It allows users to manage products, record sales and purchases, and view inventory through simple text-based menus.

- **PostgreSQL:** A powerful open-source relational database used to store all the data related to products, sales, purchases, suppliers, and employees. It ensures data integrity and supports SQL queries for managing the store's operations efficiently.

Entities, Attributes, and Descriptions

1. Employee

- **Attributes:** ID (PK), Name, Role, DOJ (Date of Joining), DOB (Date of Birth), Salary, Address
- **Description:** Employees are responsible for managing the system's operations. They record sales and handle purchase orders.

2. Customer

- **Attributes:** ID (PK), Name, Phone No
- **Description:** Customers are the buyers in the store. Each sale is linked to a customer who makes the purchase.

3. Sale

- **Attributes:** ID (PK), Date, Total Amount
- **Description:** Represents a sales transaction made by a customer, recorded by an employee. Each sale may include multiple products.

4. Sale Item (*Weak Entity*)

- **Attributes:** Quantity, Price
- **Description:** Represents individual products included in a sale. Each record links a sale with a product and stores quantity and price sold.

5. Product

- **Attributes:** ID (PK), Name, Unit Price, Cost Price, Quantity
- **Description:** Products are the items available in the store. Each product belongs to a category and is involved in both sales and purchases.

6. Category

- **Attributes:** ID (PK), Name, Description
- **Description:** Categories help in grouping products (e.g., Beverages, Snacks). One category can include multiple products.

7. Purchase Order

- **Attributes:** Order ID (PK), Date, Total Amount
- **Description:** Represents an order placed to a supplier by an employee. It consists of one or more purchase items.

8. Purchase Item (*Weak Entity*)

- **Attributes:** Quantity, Cost Price
- **Description:** Represents individual products included in a purchase order. It specifies what was ordered, how many units, and at what cost.

9. Supplier

- **Attributes:** ID (PK), Name, Phone No
- **Description:** Suppliers provide products to the store. Each purchase order is linked to a supplier.

Entity Relationships and Cardinality

1. Employee — Records — Sale

- **Cardinality:**
One **Employee** can record **many Sales**
Each **Sale** is recorded by **one Employee**
- **Type:** 1 to M (One-to-Many)
- **Description:** Employees handle customer billing, so each sale is associated with the employee who recorded it.

2. Customer — Makes — Sale

- **Cardinality:**
One **Customer** can make **many Sales**
Each **Sale** is made by **one Customer**
- **Type:** 1 to M (One-to-Many)
- **Description:** A customer may visit multiple times and make different purchases.

3. Sale — Has — Sale Item

- **Cardinality:**
One **Sale** can have **many Sale Items**
Each **Sale Item** belongs to **one Sale**
- **Type:** 1 to M (One-to-Many)
- **Description:** A sale may contain multiple products, each stored as a separate sale item entry.

4. Sale Item — Has — Product

- **Cardinality:**
Many **Sale Items** belong to **one Product**
One **Product** can be included in **many Sale Items**
- **Type:** M to 1 (Many-to-One)
- **Description:** Products appear in multiple sales, so this many-to-one link tracks quantity and price for each sale.

5. Product — Has — Category

- **Cardinality:**
Many **Products** belong to **one Category**
One **Category** can have **many Products**
- **Type:** M to 1 (Many-to-One)
- **Description:** Categories are used to group products (e.g., Beverages, Dairy).

6. Supplier — Gives — Purchase Order

- **Cardinality:**
One **Supplier** can give **many Purchase Orders**
Each **Purchase Order** is from **one Supplier**
- **Type:** 1 to M (One-to-Many)
- **Description:** The store places multiple orders to a supplier over time.

7. Employee — Manages — Purchase Order

- **Cardinality:**
One **Employee** can manage **many Purchase Orders**
Each **Purchase Order** is managed by **one Employee**
- **Type:** 1 to M (One-to-Many)
- **Description:** Employees are responsible for ordering and receiving stock.

8. Purchase Order — Consists Of — Purchase Item

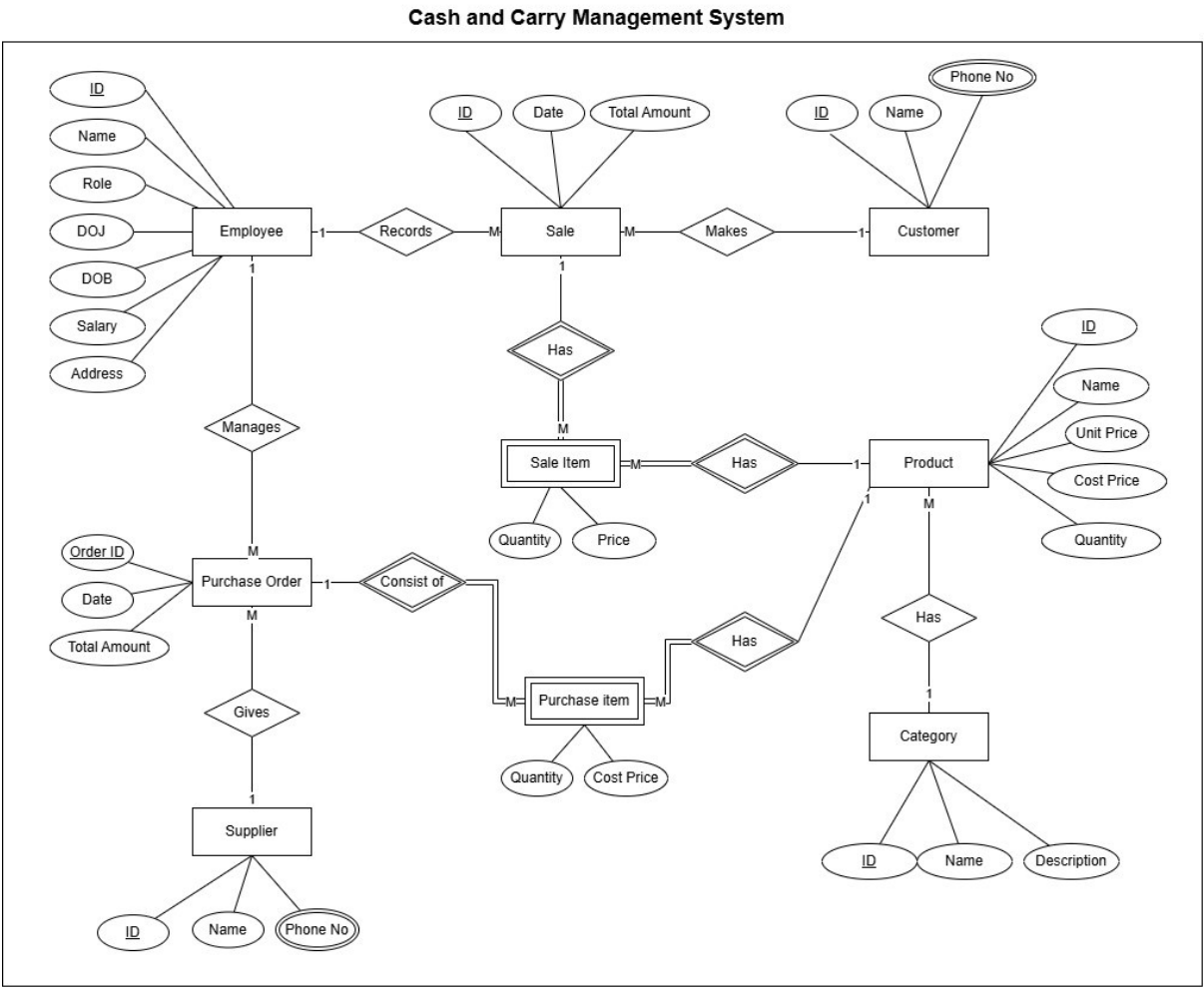
- **Cardinality:**
One **Purchase Order** can have **many Purchase Items**
Each **Purchase Item** belongs to **one Purchase Order**
- **Type:** 1 to M (One-to-Many)
- **Description:** A single purchase order may contain multiple products.

9. Purchase Item — Has — Product

- **Cardinality:**
Many **Purchase Items** refer to **one Product**
One **Product** can appear in **many Purchase Items**
- **Type:** M to 1 (Many-to-One)
- **Description:** Products are repeatedly ordered over time in different purchase orders.

Entity Relationship Diagram

Chen Model (Simple ERD):



Crow Foot Model (Relational Model):

Cash and Carry Management System

