Name: Pratham Yadav

Course: CPS 542

Batch: M3

Database Management System

Project : Inventroy Management System

Index

Table of Contents

Project Proposal: Inventory Management System

Description

The focus of my proposal is to design and develop an inventory management system database for a supermarket store. The database will allow the store manager and the business owner to keep track of all the items in their store what are their current stock quantities, who is the supplier for these items if in case the store runs out of any items from the inventory also how long does the item takes to be procured once the order is placed with the supplier. The database would hold proprietary information and is designed with the assumption that only business owners and staff working at the supermarket store are allowed to see that information.

Entities:

Items:

Items are the products that are sold by the business in the supermarket store. Attributes of Item table are Item ID, Item Name, Item Description, Item Price.

Order:

Once the items go below a particular threshold value set by the business in the inventory an order is generated to procure the item and maintain the stock levels in the inventory so that the item never goes out of stock. Attributes of Order table are Order ID, Warehouse Id, Supplier Id, Order Date.

Supplier:

A supplier in an inventory management system is a company or individual that provides goods or services to a business. By storing this information about suppliers, the inventory management system can help businesses to make better decisions about purchasing and inventory management. Attributes of Supplier table are Supplier ID, Supplier Name, Supplier Name, Supplier Contact, Street Address, City, State, Country.

Warehouse:

A warehouse is a physical location which is either owned or rented by the business i.e., the superstore in our project where all the inventory of the superstore is maintained. Each warehouse will have a unique code assigned to it so that it can be identified in the system uniquely. Attributes of Warehouse table are Warehouse ID, Warehouse Name, Street Address, City, State, Country.

Staff:

Staff is any actor who is employed by the supermarket store to carry out their day-to-day operations. The staff table will contain a unique identifier to help identify any person working for business uniquely. It will also contain their first names and last names along with their position and other contact information. Attributes of Staff table are Staff Id, Warehouse Id, First Name, Last Name, Designation, Manager Id.

Shipment:

A shipping container is a collection of goods and they are transported from one place to another. In inventory management, shipment is often associated with an order and can be used to track the movement of inventory. The shipment table can be linked to the Shipment Facilitator table, Supplier table and the warehouse table to provide additional information about the shipment. Attributes of Shipment table are Shipment Id , Warehouse Id, Supplier Id, Facilitator Id, Dispatch Address, Delivery Address, Dispatch Date.

Shipment Facilitator:

A Shipment Facilitator is someone who provides the services of delivering the shipment from one place to another i.e., from supplier to warehouse. Attributes of Shipment Facilitator table are Facilitator Id, Facilitator Name, Facilitator Cost i.e., the cost charged by the facilitator for moving the shipment from supplier to warehouse.

Transaction:

A transaction is a logical entity that is associated with inward and outward movement of items from the warehouse. Every inward and outward movement of items from the warehouse will have a unique identifier which will help the business identify the exact date and time when the event occurred along with other necessary information like warehouse name. Attributes of Transaction table are Transaction Id, Store Id, Warehouse Id, Transaction Date, Item Id, Quantity.

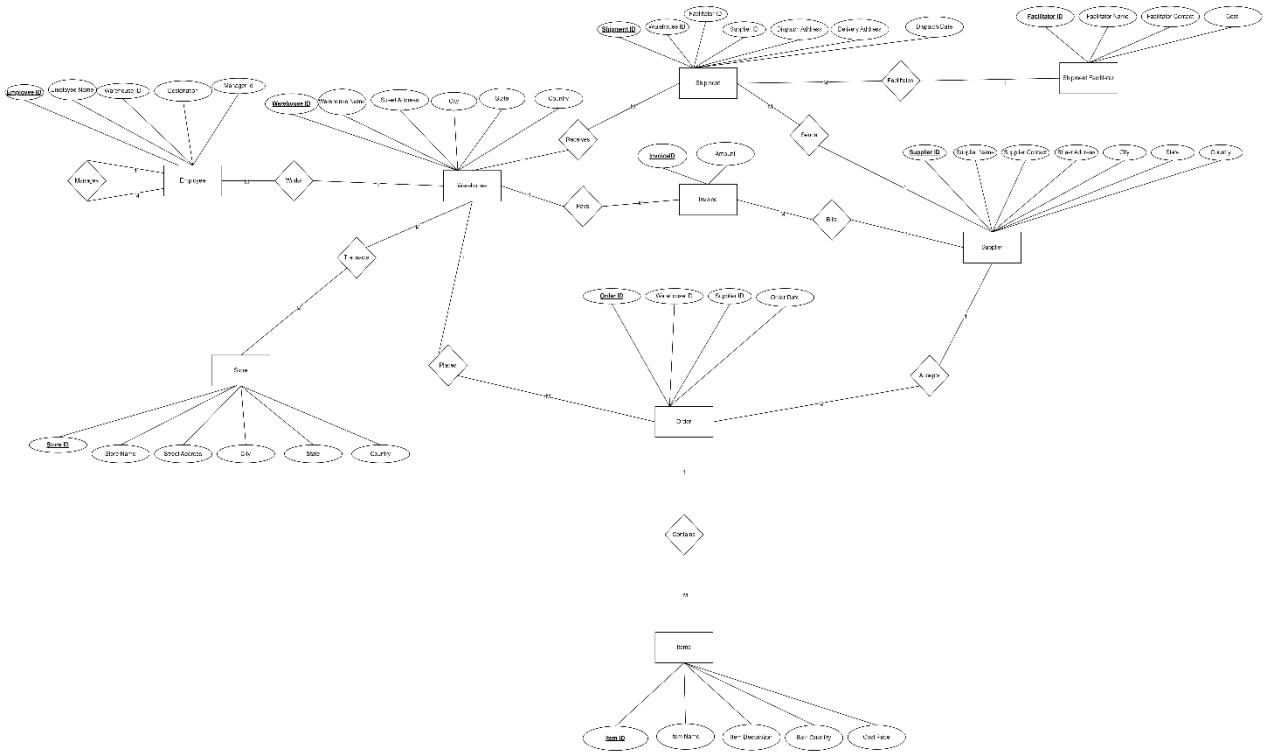
Relationships:

1. Store is linked to Transaction Table with Store ID
2. Warehouse is linked to Transaction Table with Warehouse ID
3. Warehouse is linked to Staff with Warehouse ID
4. Warehouse is linked to Shipment with Warehouse ID
5. Supplier is linked to Shipment with Supplier ID
6. Order is linked to Warehouse with Warehouse ID
7. Order is linked to Supplier with Supplier ID
8. Item is linked to Order with Order ID
9. Shipment Facilitator is Linked to Shipment table with Facilitator ID

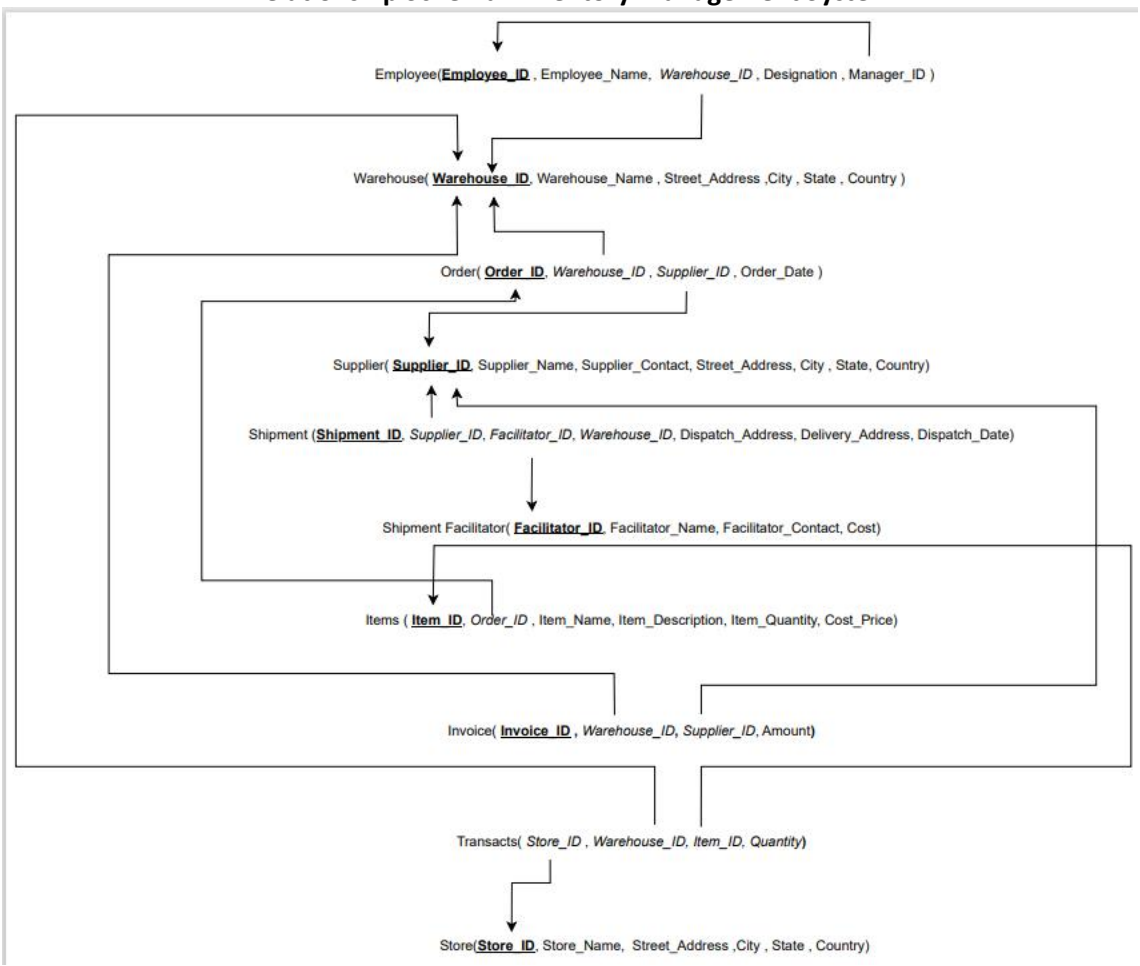
Transactions of the system:

1. Get Information about specific Employee Working in warehouse.
2. Update the details of Shipment Facilitator.
3. Delete a Shipment Facilitator from the list of Shipment Facilitators to the business.
4. Add a new warehouse to the owned by the supermarket.
5. Delete a warehouse owned by the supermarket.
6. Add a new staff member to the Inventory Management System.
7. Delete a staff member from the Inventory Management System.
8. Update the inventory of a particular item in Inventory Management System.
9. Add a new supplier to the list of suppliers to the business.
10. Delete a supplier from the list of suppliers to the business.

ER Diagram: Inventory Management System



Relationship Schema: Inventory Management System



Normalization

Table Name: Employee (Employee_ID, Employee_Name, Warehouse_ID, Designation, Manager_ID)

Candidate Key: Employee_ID
Primary Key: Employee_ID
Prime Attribute: Employee_ID
Non-Prime Attribute: Employee_Name, Warehouse_ID, Designation, Manager_ID
Functional Dependency: Employee_ID -> (Employee_Name, Warehouse_ID, Designation, Manager_ID)

First Normal Form:
We can say that the following table i.e., Employee table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Employee table only has one attribute as Candidate key i.e., Employee_ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Employee table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Employee table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Employee table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Employee table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Employee table is in BCNF

Table Name: Warehouse (Warehouse_ID, Warehouse_Name, Street_Address, City, State, Country)

Candidate Key: Warehouse _ID
Primary Key: Warehouse _ID
Prime Attribute: Warehouse _ID
Non-Prime Attribute: Warehouse_Name, Street_Address, City, State, Country
Functional Dependency: Warehouse _ID -> (Warehouse_Name, Street_Address, City, State, Country)

First Normal Form:
We can say that the following table i.e., Warehouse table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Warehouse table only has one attribute as Candidate key i.e., Warehouse _ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Warehouse table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Warehouse table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Warehouse table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Warehouse table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Warehouse table is in BCNF.

Table Name: Order (Order _ID, Warehouse_ID, Supplier_ID, Order_Date)

Candidate Key: Order _ID
Primary Key: Order _ID
Prime Attribute: Order _ID
Non-Prime Attribute: Warehouse_ID, Supplier_ID, Order_Date
Functional Dependency: Order _ID -> (Warehouse_ID, Supplier_ID, Order_Date)

First Normal Form:
We can say that the following table i.e., Order table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Order table only has one attribute as Candidate key i.e., Order _ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Order table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Order table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Order table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Order table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Order table is in BCNF.

Table Name: Supplier (Supplier_ID, Supplier_Name, Supplier_Contact, Street_Address, City, State, Country)

Candidate Key: Supplier _ID
Primary Key: Supplier _ID
Prime Attribute: Supplier _ID
Non-Prime Attribute: Supplier_Name, Supplier_Contact, Street_Address, City, State, Country
Functional Dependency: Supplier _ID-> (Supplier_Name, Supplier_Contact, Street_Address, City, State, Country)

First Normal Form:
We can say that the following table i.e., Supplier table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Supplier table only has one attribute as Candidate key i.e., Supplier _ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Supplier table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Supplier table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Supplier table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Supplier table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Supplier table is in BCNF.

Table Name: Shipment (Shipment _ID, Supplier_ID, Facilitator_ID, Warehouse_ID, Dispatch_Address, Delivery_Address Dispatch_Date)

Candidate Key: Shipment _ID
Primary Key: Shipment _ID
Prime Attribute: Shipment _ID
Non-Prime Attribute: Supplier_ID, Facilitator_ID, Warehouse_ID, Dispatch_Address, Delivery_Address Dispatch_Date
Functional Dependency: Shipment _ID -> (Supplier_ID, Facilitator_ID, Warehouse_ID, Dispatch_Address, Delivery_Address Dispatch_Date)

First Normal Form:
We can say that the following table i.e., Shipment table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Shipment table only has one attribute as Candidate key i.e., Shipment _ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Shipment table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Shipment table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Shipment table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Shipment table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Shipment table is in BCNF.

Table Name: Shipment Facilitator (Facilitator _ID, Facilitator_Name, Facilitator_Contact ,Cost)

Candidate Key: Facilitator _ID
Primary Key: Facilitator _ID
Prime Attribute: Facilitator _ID
Non-Prime Attribute: Facilitator_Name, Facilitator_Contact , Cost
Functional Dependency: Facilitator _ID-> (Facilitator_Name, Facilitator_Contact , Cost)

First Normal Form:
We can say that the following table i.e., Shipment Facilitator table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Shipment Facilitator table only has one attribute as Candidate key i.e., Facilitator _ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Shipment Facilitator table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Shipment Facilitator table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Shipment Facilitator table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Shipment Facilitator table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Shipment Facilitator table is in BCNF.

Table Name: Items (Item_ID, Order_ID, Item_Name, Item_Description ,Item_Quantity, Cost_Price)

Candidate Key: Item _ID
Primary Key: Item _ID
Prime Attribute: Item_ID
Non-Prime Attribute: Order_ID, Item_Name, Item_Description ,Item_Quantity, Cost_Price
Functional Dependency: Item _ID -> (Order_ID, Item_Name, Item_Description ,Item_Quantity, Cost_Price)

First Normal Form:
We can say that the following table i.e., Items table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Items table only has one attribute as Candidate key i.e., Item_ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Items table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Items table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Items table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Items table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Items table is in BCNF.

Table Name: Invoice (Invoice_ID, Warehouse_ID, Supplier_ID, Amount)

Candidate Key: Invoice _ID
Primary Key: Invoice _ID
Prime Attribute: Invoice _ID
Non-Prime Attribute: Warehouse_ID, Supplier_ID, Amount
Functional Dependency: Invoice _ID-> (Warehouse_ID, Supplier_ID, Amount)

First Normal Form:
We can say that the following table i.e., Invoice table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Invoice table only has one attribute as Candidate key i.e., Invoice_ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Invoice table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Invoice table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Invoice table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Invoice table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Invoice table is in BCNF.

Table Name: Transacts (Store_ID, Warehouse_ID, Item_ID, Quantity,Transaction_Date)

Candidate Key: Store_ID, Warehouse_ID, Item_ID
Primary Key: Store_ID, Warehouse_ID, Item_ID
Prime Attribute: Store_ID, Warehouse_ID, Item_ID
Non-Prime Attribute: Quantity, Transaction_Date
Functional Dependency: Store_ID, Warehouse_ID, Item_ID-> (Quantity, Transaction_Date)

First Normal Form:
We can say that the following table i.e., Transacts table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Transacts table has three attributes as Candidate key i.e., Store_ID, Warehouse_ID, Item_ID and other attribute i.e., Quantity of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Transacts table is in second normal form, Since Quantity Attribute of the table is fully functionally dependent on Store_ID, Warehouse_ID, Item_ID.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

The Invoice table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Transacts table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Transacts table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Transacts table is in BCNF.

Table Name: Store (Store_ID, Store _Name, Street_Address, City, State, Country)

Candidate Key: Store _ID
Primary Key: Store _ID
Prime Attribute: Store _ID
Non-Prime Attribute: Store _Name, Street_Address, City, State, Country
Functional Dependency: Store _ID-> (Store _Name, Street_Address, City, State, Country)

First Normal Form:
We can say that the following table i.e., Store table is in first normal form because each cell of table contains single value.

Second Normal Form:
According to second normal form, we can say that the table is in second normal form if and only if the table satisfies two conditions

- Table under consideration must be in First Normal Form
- There must be no presence of partial dependency i.e., any attribute of the table instead of depending on the entire candidate key depends on the part of candidate key.

The Store table only has one attribute as Candidate key i.e., Store _ID and all other attributes of the table are determined from the following candidate key and we have already proved that the table is in first normal form thus we can conclude that the Store table is in second normal form.

Third Normal Form:
According to third normal form, we can say that the table is in third normal form if and only if the table satisfies two conditions

- Table under consideration must be in Second Normal Form
- There must be no presence of Transitive Dependency i.e., one non-prime attribute must not determine another non-prime attribute.

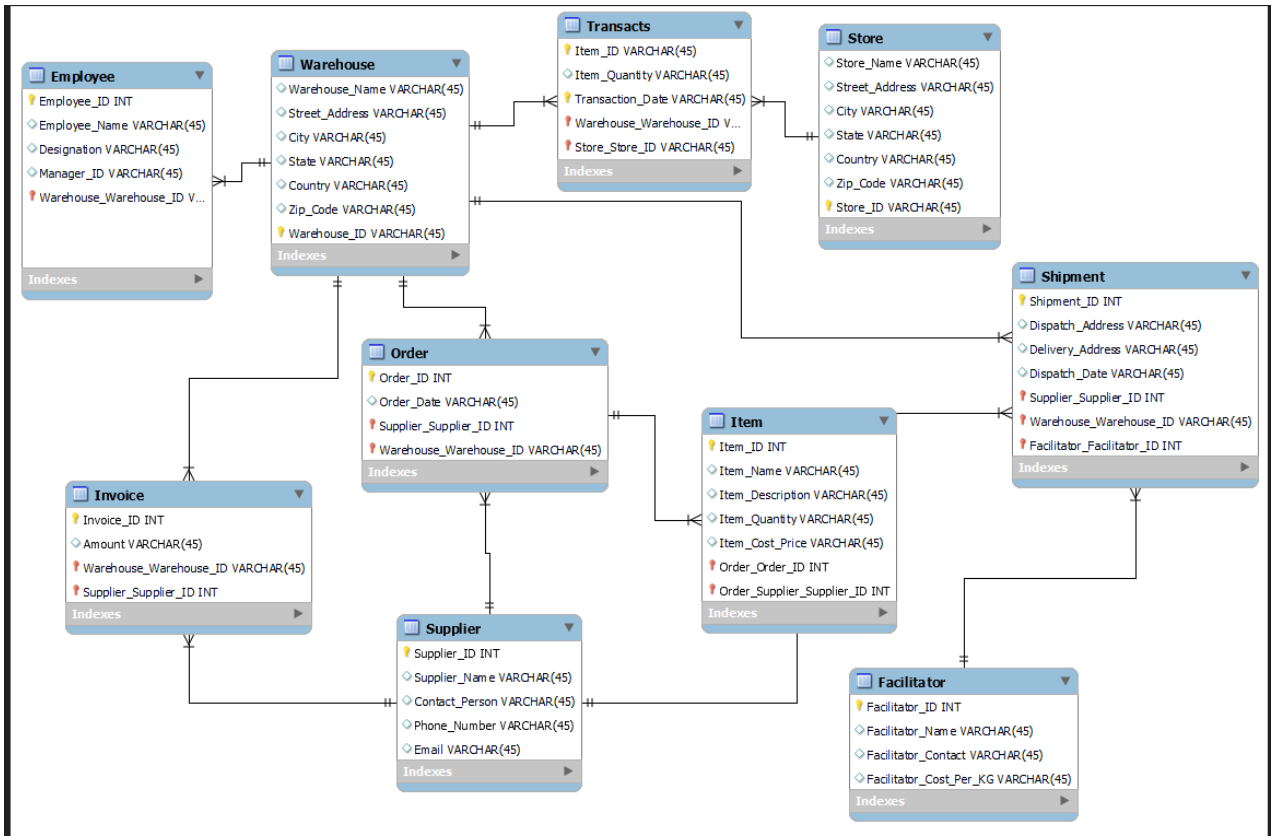
The Store table has no such scenarios in which a non-prime attribute i.e., the attribute which is not a part of candidate key determines another non-prime attribute. Thus, we can conclude that the Store table is in Third Normal Form.

Boyce Codd Normal Form:
According to Boyce Codd Normal Form or Fourth Normal Form a table is said to be in BCNF if and only if it satisfies the following conditions

- The table under consideration must be in Third Normal Form
- There must be no such scenarios wherein a non-prime or a prime attribute of the table determines a prime attribute.

The Store table has no such scenarios in which a non-prime or a prime attribute determines another prime attribute. Thus, we can conclude that the Store table is in BCNF.

Logical Model



Physical Model

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FO
R_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema Inventory_Management_System
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema Inventory_Management_System
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `Inventory_Management_System` DEFAULT CHARACTER SET utf8 ;
USE `Inventory_Management_System` ;

-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Warehouse`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Warehouse` (
  `Warehouse_Name` VARCHAR(45) NULL,
  `Street_Address` VARCHAR(45) NULL,
  `City` VARCHAR(45) NULL,
  `State` VARCHAR(45) NULL,
  `Country` VARCHAR(45) NULL,
  `Zip_Code` VARCHAR(45) NULL,
  `Warehouse_ID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Warehouse_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Employee` (
  `Employee_ID` INT NOT NULL,
  `Employee_Name` VARCHAR(45) NULL,
  `Designation` VARCHAR(45) NULL,
  `Manager_ID` VARCHAR(45) NULL,
  `Warehouse_Warehouse_ID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Employee_ID`, `Warehouse_Warehouse_ID`),
  INDEX `fk_Employee_Warehouse1_idx` (`Warehouse_Warehouse_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Employee_Warehouse1`
    FOREIGN KEY (`Warehouse_Warehouse_ID`)
    REFERENCES `Inventory_Management_System`.`Warehouse` (`Warehouse_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Supplier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Supplier` (
  `Supplier_ID` INT NOT NULL,
  `Supplier_Name` VARCHAR(45) NULL,
  `Contact_Person` VARCHAR(45) NULL,
  `Phone_Number` VARCHAR(45) NULL,
  `Email` VARCHAR(45) NULL,
  PRIMARY KEY (`Supplier_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Order` (
  `Order_ID` INT NOT NULL,
  `Order_Date` VARCHAR(45) NULL,
  `Supplier_Supplier_ID` INT NOT NULL,
  `Warehouse_Warehouse_ID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Order_ID`, `Supplier_Supplier_ID`, `Warehouse_Warehouse_ID`),
  INDEX `fk_Order_Supplier1_idx` (`Supplier_Supplier_ID` ASC) VISIBLE,
  INDEX `fk_Order_Warehouse1_idx` (`Warehouse_Warehouse_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Order_Supplier1`
    FOREIGN KEY (`Supplier_Supplier_ID`)
    REFERENCES `Inventory_Management_System`.`Supplier` (`Supplier_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Order_Warehouse1`
    FOREIGN KEY (`Warehouse_Warehouse_ID`)
    REFERENCES `Inventory_Management_System`.`Warehouse` (`Warehouse_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Facilitator`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Facilitator` (
  `Facilitator_ID` INT NOT NULL,
  `Facilitator_Name` VARCHAR(45) NULL,
  `Facilitator_Contact` VARCHAR(45) NULL,
  `Facilitator_Cost_Per_KG` VARCHAR(45) NULL,
  PRIMARY KEY (`Facilitator_ID`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Shipment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Shipment` (
  `Shipment_ID` INT NOT NULL,
  `Dispatch_Address` VARCHAR(45) NULL,
  `Delivery_Address` VARCHAR(45) NULL,
  `Dispatch_Date` VARCHAR(45) NULL,
  `Supplier_Supplier_ID` INT NOT NULL,
  `Warehouse_Warehouse_ID` VARCHAR(45) NOT NULL,
  `Facilitator_Facilitator_ID` INT NOT NULL,
  PRIMARY KEY (`Shipment_ID`, `Supplier_Supplier_ID`, `Warehouse_Warehouse_ID`,
`Facilitator_Facilitator_ID`),
  INDEX `fk_Shipment_Supplier1_idx` (`Supplier_Supplier_ID` ASC) VISIBLE,
  INDEX `fk_Shipment_Warehouse1_idx` (`Warehouse_Warehouse_ID` ASC) VISIBLE,
  INDEX `fk_Shipment_Shipment Facilitator1_idx` (`Facilitator_Facilitator_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Shipment_Supplier1`
    FOREIGN KEY (`Supplier_Supplier_ID`)
    REFERENCES `Inventory_Management_System`.`Supplier` (`Supplier_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Shipment_Warehouse1`
    FOREIGN KEY (`Warehouse_Warehouse_ID`)
    REFERENCES `Inventory_Management_System`.`Warehouse` (`Warehouse_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Shipment_Shipment Facilitator1`
    FOREIGN KEY (`Facilitator_Facilitator_ID`)
    REFERENCES `Inventory_Management_System`.`Facilitator` (`Facilitator_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Item`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Item` (
  `Item_ID` INT NOT NULL,
  `Item_Name` VARCHAR(45) NULL,
  `Item_Description` VARCHAR(45) NULL,
  `Item_Quantity` VARCHAR(45) NULL,
  `Item_Cost_Price` VARCHAR(45) NULL,
  `Order_Order_ID` INT NOT NULL,
  `Order_Supplier_Supplier_ID` INT NOT NULL,
  PRIMARY KEY (`Item_ID`, `Order_Order_ID`, `Order_Supplier_Supplier_ID`),
  INDEX `fk_Item_Order1_idx` (`Order_Order_ID` ASC, `Order_Supplier_Supplier_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Item_Order1`
    FOREIGN KEY (`Order_Order_ID` , `Order_Supplier_Supplier_ID`)
    REFERENCES `Inventory_Management_System`.`Order` (`Order_ID` , `Supplier_Supplier_ID`)
```

```
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Invoice`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Invoice` (
  `Invoice_ID` INT NOT NULL,
  `Amount` VARCHAR(45) NULL,
  `Warehouse_Warehouse_ID` VARCHAR(45) NOT NULL,
  `Supplier_Supplier_ID` INT NOT NULL,
  PRIMARY KEY (`Invoice_ID`, `Warehouse_Warehouse_ID`, `Supplier_Supplier_ID`),
  INDEX `fk_Invoice_Warehouse1_idx` (`Warehouse_Warehouse_ID` ASC) VISIBLE,
  INDEX `fk_Invoice_Supplier1_idx` (`Supplier_Supplier_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Invoice_Warehouse1`
    FOREIGN KEY (`Warehouse_Warehouse_ID`)
    REFERENCES `Inventory_Management_System`.`Warehouse` (`Warehouse_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Invoice_Supplier1`
    FOREIGN KEY (`Supplier_Supplier_ID`)
    REFERENCES `Inventory_Management_System`.`Supplier` (`Supplier_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Store`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Store` (
  `Store_Name` VARCHAR(45) NULL,
  `Street_Address` VARCHAR(45) NULL,
  `City` VARCHAR(45) NULL,
  `State` VARCHAR(45) NULL,
  `Country` VARCHAR(45) NULL,
  `Zip_Code` VARCHAR(45) NULL,
  `Store_ID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Store_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Inventory_Management_System`.`Transacts`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Inventory_Management_System`.`Transacts` (
  `Item_ID` VARCHAR(45) NOT NULL,
  `Item_Quantity` VARCHAR(45) NULL,
  `Transaction_Date` VARCHAR(45) NOT NULL,
```

```
  `Warehouse_Warehouse_ID` VARCHAR(45) NOT NULL,
  `Store_Store_ID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Item_ID`, `Transaction_Date`, `Warehouse_Warehouse_ID`, `Store_Store_ID`),
  INDEX `fk_Transacts_Warehouse1_idx` (`Warehouse_Warehouse_ID` ASC) VISIBLE,
  INDEX `fk_Transacts_Store1_idx` (`Store_Store_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Transacts_Warehouse1`
    FOREIGN KEY (`Warehouse_Warehouse_ID`)
    REFERENCES `Inventory_Management_System`.`Warehouse` (`Warehouse_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Transacts_Store1`
    FOREIGN KEY (`Store_Store_ID`)
    REFERENCES `Inventory_Management_System`.`Store` (`Store_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Transactions

Warehouse Table:

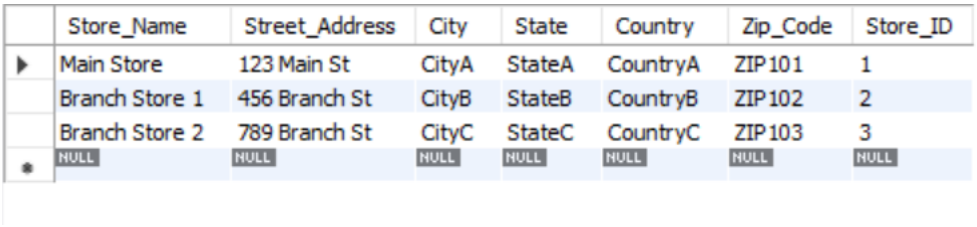
```
INSERT INTO Warehouse (Warehouse_ID, Warehouse_Name, Street_Address, City, State, Country)
VALUES
   (101, 'Main Warehouse', '123 Main St', 'CityA', 'StateA', 'CountryA'),
   (102, 'Secondary Warehouse', '456 Second St', 'CityB', 'StateB', 'CountryB'),
   (103, 'Regional Warehouse', '789 Third St', 'CityC', 'StateC', 'CountryC'),
   (104, 'Distribution Center', '101 DC St', 'CityD', 'StateD', 'CountryD');


--Delete
DELETE FROM Warehouse WHERE Warehouse_ID = 101;


--View the table
SELECT * FROM warehouse;
```

Warehouse_Name	Street_Address	City	State	Country	Zip_Code	Warehouse_ID
Main Warehouse	123 Main St	CityA	StateA	CountryA	ZIP101	101
Secondary Warehouse	456 Second St	CityB	StateB	CountryB	ZIP102	102
Regional Warehouse	789 Third St	CityC	StateC	CountryC	ZIP103	103
Distribution Center	101 DC St	CityD	StateD	CountryD	ZIP104	104
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Store Table:

```
INSERT INTO Store (Store_ID, Store_Name, Street_Address, City, State, Country,Zip_Code)
VALUES
   (1, 'Main Store', '123 Main St', 'CityA', 'StateA', 'CountryA','ZIP101'),
   (2, 'Branch Store 1', '456 Branch St', 'CityB', 'StateB', 'CountryB','ZIP102'),
   (3, 'Branch Store 2', '789 Branch St', 'CityC', 'StateC', 'CountryC','ZIP103');
```

```
Select * from Store;
```



Store_Name	Street_Address	City	State	Country	Zip_Code	Store_ID
Main Store	123 Main St	CityA	StateA	CountryA	ZIP101	1
Branch Store 1	456 Branch St	CityB	StateB	CountryB	ZIP102	2
Branch Store 2	789 Branch St	CityC	StateC	CountryC	ZIP103	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Transacts Table:

-- Insert sample data into the Transacts table

INSERT INTO Transacts (Store_Store_ID, Warehouse_Warehouse_ID, Item_ID, Item_Quantity, Transaction_Date)

VALUES

(1, 101, 1, 50, '2023-01-20'),

(2, 102, 2, 25, '2023-02-15'),

(1, 101, 3, 30, '2023-03-10'),

(3, 103, 4, 15, '2023-04-05');

SELECT * FROM TRANSACTS;

Item_ID	Item_Quantity	Transaction_Date	Warehouse_Warehouse_ID	Store_Store_ID
1	50	2023-01-20	101	1
2	25	2023-02-15	102	2
3	30	2023-03-10	101	1
4	15	2023-04-05	103	3
NULL	NULL	NULL	NULL	NULL

Shipment Facilitator:

-- Insert sample data into the Shipment_Facilitator table

INSERT INTO Facilitator (Facilitator_ID, Facilitator_Name, Facilitator_Contact, Facilitator_Cost_Per_KG)

VALUES

(301, 'Express Logistics', '123-456-7890', 1.50),

(302, 'Swift Shippers', '987-654-3210', 1.75),

(303, 'Global Transporters', '111-222-3333', 2.00),

(304, 'Speedy Movers', '444-555-6666', 2.25);

--Update

UPDATE Shipment_Facilitator SET Cost_Per_KG = 2.5WHERE Facilitator_ID = 301;

--Delete

DELETE FROM Shipment_Facilitator WHERE Facilitator_ID =301;

-- View the table

Select * FROM Facilitator;

Facilitator_ID	Facilitator_Name	Facilitator_Contact	Facilitator_Cost_Per_KG
301	Express Logistics	123-456-7890	1.50
302	Swift Shippers	987-654-3210	1.75
303	Global Transporters	111-222-3333	2.00
304	Speedy Movers	444-555-6666	2.25
NULL	NULL	NULL	NULL

Shipment Table:

-- Insert sample data into the Shipment table

INSERT INTO Shipment (Shipment_ID, Supplier_Supplier_ID, Facilitator_Facilitator_ID, Warehouse_Warehouse_ID, Dispatch_Address, Delivery_Address, Dispatch_Date)

VALUES

(1, 201, 301, 101, '123 Supplier St, CityA, StateA', '456 Warehouse St, CityA, StateA', '2023-01-20'),

(2, 202, 302, 102, '456 Supplier St, CityB, StateB', '789 Warehouse St, CityB, StateB', '2023-02-15'),

(3, 203, 303, 103, '789 Supplier St, CityC, StateC', '101 Warehouse St, CityC, StateC', '2023-03-10'),

(4, 204, 304, 104, '101 Supplier St, CityD, StateD', '1234 Warehouse St, CityD, StateD', '2023-04-05');

SELECT * FROM SHIPMENT;

Shipment_ID	Dispatch_Address	Delivery_Address	Dispatch_Date	Supplier_Supplier_ID	Warehouse_Warehous
1	123 Supplier St, CityA, StateA	456 Warehouse St, CityA, StateA	2023-01-20	201	101
2	456 Supplier St, CityB, StateB	789 Warehouse St, CityB, StateB	2023-02-15	202	102
3	789 Supplier St, CityC, StateC	101 Warehouse St, CityC, StateC	2023-03-10	203	103
4	101 Supplier St, CityD, StateD	1234 Warehouse St, CityD, StateD	2023-04-05	204	104
NULL	NULL	NULL	NULL	NULL	NULL

Supplier Table:

```
-- Insert sample data into the Supplier table

INSERT INTO Supplier (Supplier_ID, Supplier_Name, Contact_Person, Phone_Number, Email)

VALUES

   (201, 'ABC Suppliers', 'John Supplier', '123-456-7890', 'john@example.com'),

   (202, 'XYZ Distributors', 'Jane Distributor', '987-654-3210', 'jane@example.com'),

   (203, 'LMN Imports', 'Mike Importer', '111-222-3333', 'mike@example.com'),

   (204, 'PQR Exports', 'Sara Exporter', '444-555-6666', 'sara@example.com');


-- Delete a supplier

DELETE FROM Supplier WHERE Supplier_ID = 204;


-- View the table

SELECT * FROM Supplier;
```

Supplier_ID	Supplier_Name	Contact_Person	Phone_Number	Email
201	ABC Suppliers	John Supplier	123-456-7890	john@example.com
202	XYZ Distributors	Jane Distributor	987-654-3210	jane@example.com
203	LMN Imports	Mike Importer	111-222-3333	mike@example.com
204	PQR Exports	Sara Exporter	444-555-6666	sara@example.com
NULL	NULL	NULL	NULL	NULL

Item Table:

-- Insert sample data into the Items table

INSERT INTO Item (Item_ID, Order_Order_ID, Item_Name, Item_Description, Item_Quantity, Item_Cost_Price,Order_Supplier_Supplier_ID)

VALUES

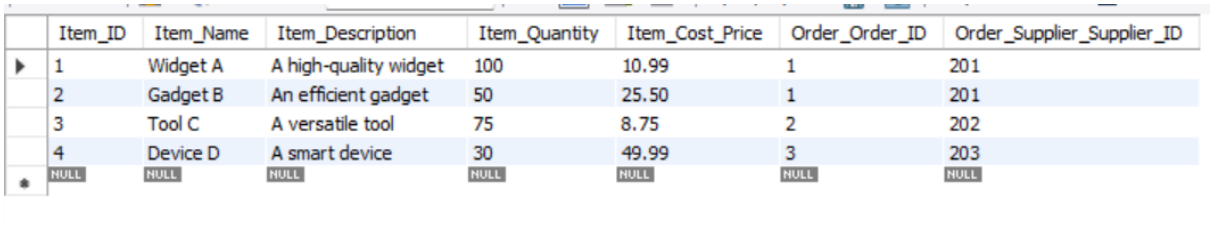
(1, 1, 'Widget A', 'A high-quality widget', 100, 10.99,201),

(2, 1, 'Gadget B', 'An efficient gadget', 50, 25.50,201),

(3, 2, 'Tool C', 'A versatile tool', 75, 8.75,202),

(4, 3, 'Device D', 'A smart device', 30, 49.99,203);

SELECT * FROM Item;



Item_ID	Item_Name	Item_Description	Item_Quantity	Item_Cost_Price	Order_Order_ID	Order_Supplier_Supplier_ID
1	Widget A	A high-quality widget	100	10.99	1	201
2	Gadget B	An efficient gadget	50	25.50	1	201
3	Tool C	A versatile tool	75	8.75	2	202
4	Device D	A smart device	30	49.99	3	203
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Order Table:

-- Insert sample data into the Order table

INSERT INTO inventory_management_system.ORDER (Order_ID, Warehouse_Warehouse_ID, Supplier_Supplier_ID, Order_Date)

VALUES

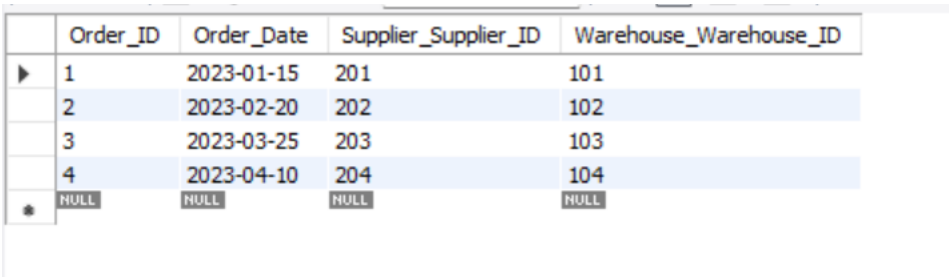
(1, 101, 201, '2023-01-15'),

(2, 102, 202, '2023-02-20'),

(3, 103, 203, '2023-03-25'),

(4, 104, 204, '2023-04-10');

SELECT * FROM inventory_management_system.ORDER;



Order_ID	Order_Date	Supplier_Supplier_ID	Warehouse_Warehouse_ID
1	2023-01-15	201	101
2	2023-02-20	202	102
3	2023-03-25	203	103
4	2023-04-10	204	104
NULL	NULL	NULL	NULL

Invoice Table:

-- Insert sample data into the Invoice table

INSERT INTO Invoice (Invoice_ID, Warehouse_Warehouse_ID, Supplier_Supplier_ID, Amount)

VALUES

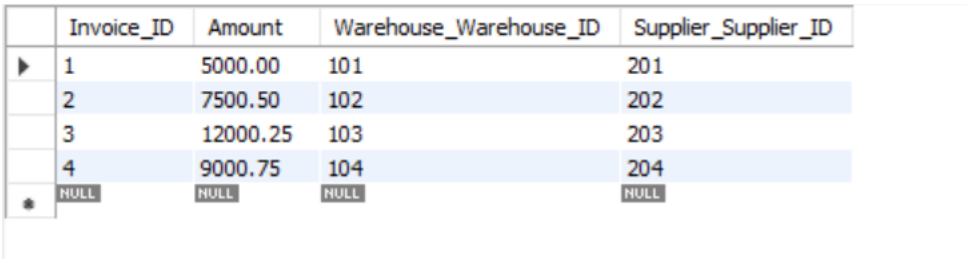
(1, 101, 201, 5000.00),

(2, 102, 202, 7500.50),

(3, 103, 203, 12000.25),

(4, 104, 204, 9000.75);

SELECT * FROM INVOICE;



Invoice_ID	Amount	Warehouse_Warehouse_ID	Supplier_Supplier_ID
1	5000.00	101	201
2	7500.50	102	202
3	12000.25	103	203
4	9000.75	104	204
NULL	NULL	NULL	NULL

Employee Table

```
INSERT INTO Employee (Employee_ID, Employee_Name, Warehouse_ID, Designation, Manager_ID)
VALUES
  (1, 'John Doe', 101, 'Manager', NULL),
  (2, 'Jane Smith', 101, 'Supervisor', 1),
  (3, 'Bob Johnson', 102, 'Clerk', 2),
  (4, 'Alice White', 102, 'Clerk', 2),
  (5, 'Tom Brown', 103, 'Supervisor', 1),
  (6, 'Sarah Green', 103, 'Clerk', 5),
  (7, 'Mark Davis', 104, 'Manager', NULL),
  (8, 'Emily Wilson', 104, 'Supervisor', 7),
  (9, 'Chris Lee', 104, 'Clerk', 8);


--Delete an Employee
DELETE FROM Employee WHERE Employee_ID =9;


--View the table
SELECT * FROM Employee
```

Employee_ID	Employee_Name	Designation	Manager_ID	Warehouse_Warehouse_ID
1	John Doe	Manager	NULL	101
2	Jane Smith	Supervisor	1	101
3	Bob Johnson	Clerk	2	102
4	Alice White	Clerk	2	102
5	Tom Brown	Supervisor	1	103
6	Sarah Green	Clerk	5	103
7	Mark Davis	Manager	NULL	104

Conclusion

The focus of this project was to design and develop an inventory management system database for a supermarket store. The database allows the store manager and business owner to track suppliers, orders, shipments, and transactions.

Key entities in the database include Items, Orders, Suppliers, Warehouses, Shipments, Shipment Facilitators, Staff, Transactions, Stores, and Invoices. These entities and their attributes were defined to capture all the necessary data points to effectively manage store inventory.

Relationships were established between the entities to link the data and enable useful reporting for planning and analysis. For example, Orders are linked to Suppliers and Warehouses, Shipments are linked to Warehouses and Suppliers, and Transactions are linked to Stores, Warehouses, and Items.

The database structure was normalized to BCNF form to eliminate data redundancy and ensure data integrity. Functional dependencies were analysed and the tables were assessed to satisfy conditions for the appropriate normal forms.

Through this project I gained hands-on experience designing an end-to-end database solution for a business use case. I learned database concepts such as entities, attributes, relationships, normalization, and transactions. I also improved technical skills in diagramming ERDs, defining table schemas, and using a database management system.

The completed inventory management database provides the supermarket with a valuable tool to gain visibility into their inventory operations. Key features like automatic reorder points and transaction logging streamline supply chain and warehouse workflows. Overall, this project sharpened my database design abilities while producing a functional system that solves real business needs.