**Name: Pratham Yadav**

**Course: CPS 542**

**Batch: M3**

**Database Management System**

**Project : Inventroy Management System**

Index

Table of Contents

**Project Proposal3**

Description3

Relationships4

List of Transactions5

**Entity Relationship Diagram6**

**Relationship Schema7**

**Normalization8**

**Logical Model18**

**Physical Model17**

**Transaction Queries24**

**Conclusion34**

# 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

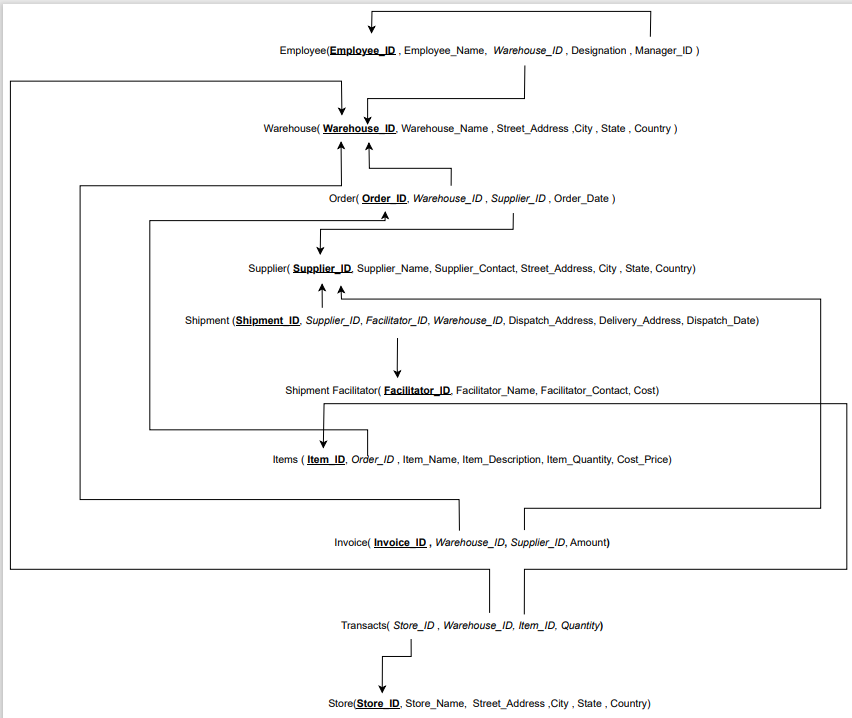
# Transactions of the system:

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.

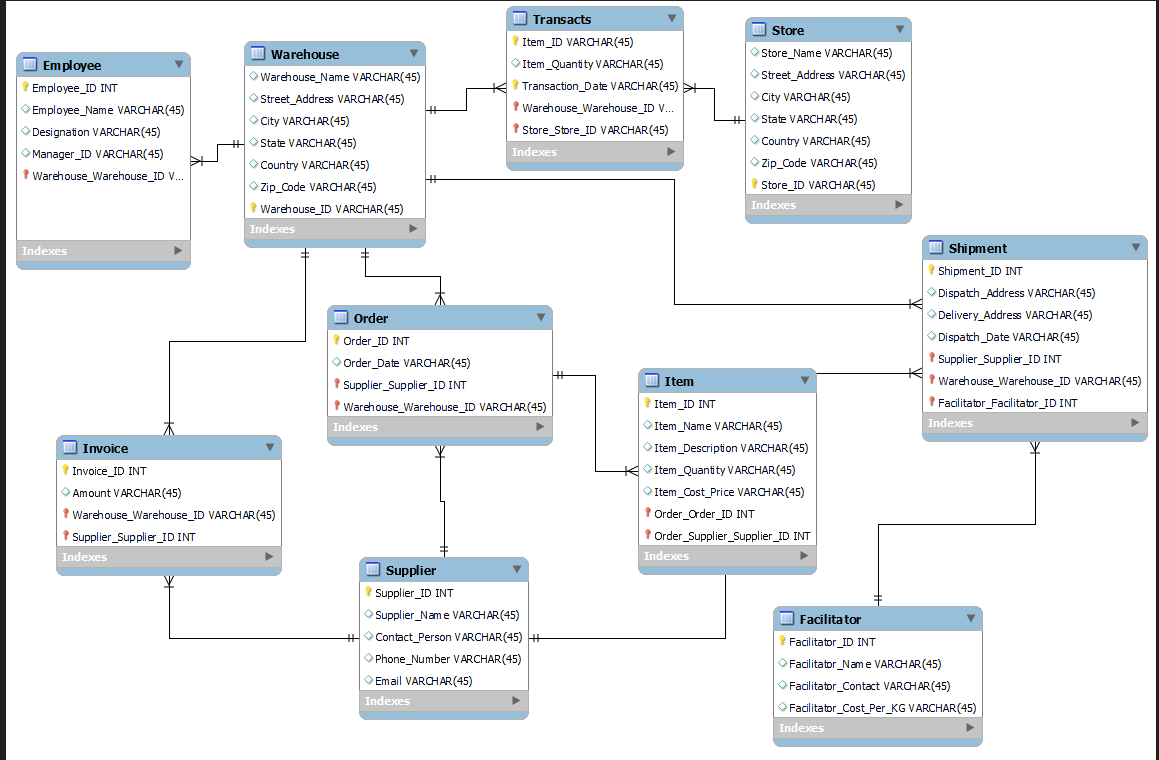
**Boyce Codd Normal Form:**

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\_FOR\_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;

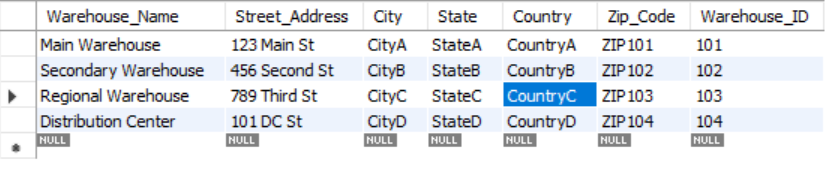
SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

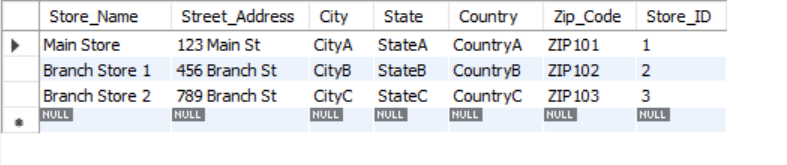
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

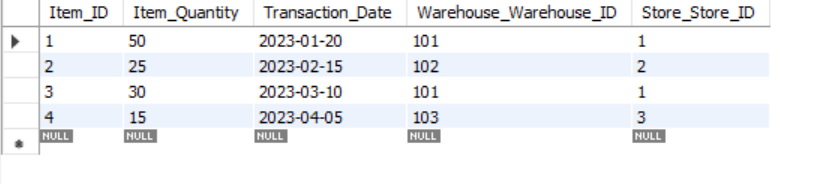
**Transactions**

**Warehouse Table:**



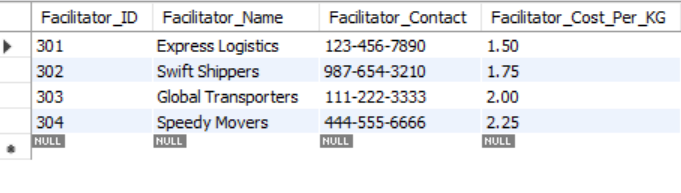
**Store Table:**

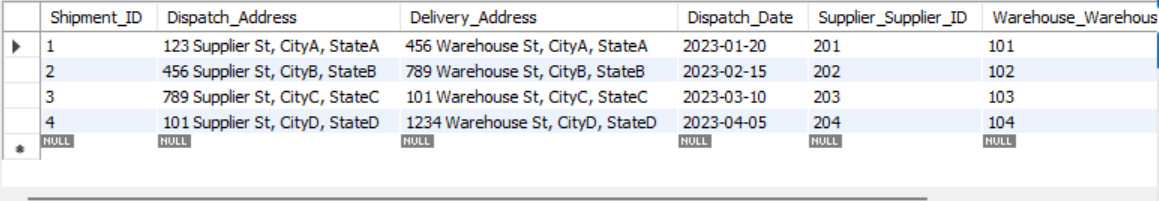


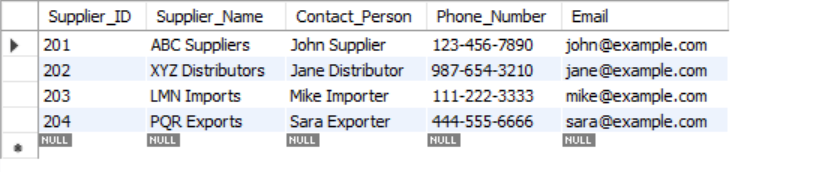
**Transacts Table:**

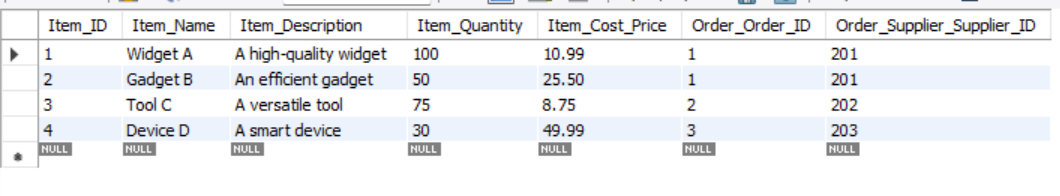
**Shipment Facilitator:**

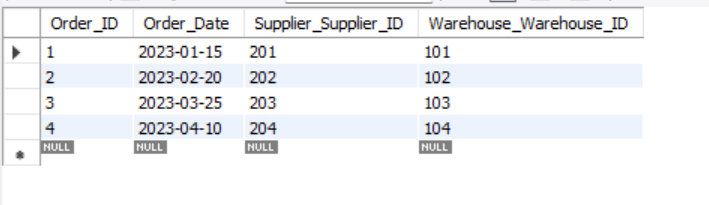




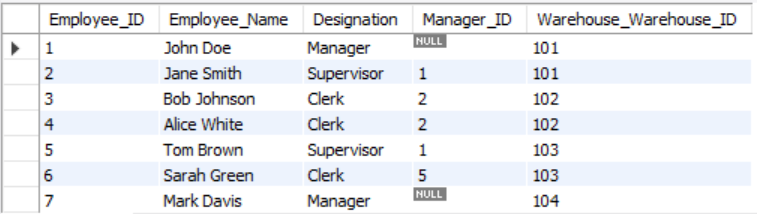
**Shipment Table:**

**Supplier Table:**

**Item Table:**

**Order Table:**

**Invoice Table:**

**Employee Table**

**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.