

Southeast University

Department Of Computer Science & Engineering



Course Code : CSE384

Course Name : Database Design

Project Name : Inventory Management System

Date : 27-08-2022

Submitted By

Most. Laila Akter

2019000010096

Submitted To

Mr. Nahin Kumar Dey

Lecturer, Southeast University

Project Overview:

Inventory management is the process of ordering, storing, using, and selling a company's inventory. This includes the management of raw materials, components, finished products, warehousing and processing of such items. In manufacturing enterprises, managing inventory is a difficult task. Many challenges arise in supply chains. So, Inventory management system needs a database to store these data on these products, details & customers.

This inventory management system will help to track the correct inventory in short time and it will be easier to manage and sort out the inventory as well. These will help to manage the products, order rate and employees.

The system will also have customer & supplier. Customers can choose what products they will be placing order from the inventory and supplier will be able to provide stocks.

Inventory management system has seven entities. They are:

DB Table	Description
Brand/inventory	This has details of inventory id, product id, quantity and its updated time.
Products	It has details of product id, slugs, name, price, vat and references.
Item	Details of a specific item name, id ,update date etc
order	It has information on order id, product id, arrival time, order time, deadline etc.
User	This is user information on his id, name, phone, address and DOB.
Order_items	It has details of customer id, order time and ordered time
Product_meta	This has product id, order id, ordering time, price & vat.

Functionality:

Database allows admins to do:

1. Enter customer information and place their orders. Each product has its own order details.
2. Enter user services by supplying the products. Each products has its own product meta.
3. Enter products information and place the products by their suitable inventory

Some procedure & function for the database are:

1. Trigger (Price): Products should be price included.
2. Trigger (Vat): Product should be vat included.
3. Function (slug): to identify products within short time

Targeted Audience:

This database is meant to be used by the companies which have shown multiple problems arranging their system, sorting out the products, getting orders from the users and problem stocks. So, if they integrate this database, they solve their storing problems. Many big industries like Food, Cloth, hardware etc. face multiple inventory issues. So, it can be effective for them

Database SQL:

User:

```
CREATE TABLE `inventory`.`user` (  
  `id` BIGINT NOT NULL AUTO_INCREMENT,  
  `roleId` SMALLINT NOT NULL,  
  `firstName` VARCHAR(50) NULL DEFAULT NULL,  
  `middleName` VARCHAR(50) NULL DEFAULT NULL,  
  `lastName` VARCHAR(50) NULL DEFAULT NULL,  
  `username` VARCHAR(50) NULL DEFAULT NULL,  
  `mobile` VARCHAR(15) NULL,  
  `email` VARCHAR(50) NULL,  
  `passwordHash` VARCHAR(32) NOT NULL,  
  `registeredAt` DATETIME NOT NULL,  
  `lastLogin` DATETIME NULL DEFAULT NULL,  
  `intro` TINYTEXT NULL DEFAULT NULL,  
  `profile` TEXT NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),
```

```
UNIQUE INDEX `uq_username` (`username` ASC),
UNIQUE INDEX `uq_mobile` (`mobile` ASC),
UNIQUE INDEX `uq_email` (`email` ASC );
```

Product:

```
CREATE TABLE `inventory`.`product` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `title` VARCHAR(75) NOT NULL,
  `summary` TINYTEXT NULL,
  `type` SMALLINT(6) NOT NULL DEFAULT 0,
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NULL DEFAULT NULL,
  `content` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

Product meta

```
CREATE TABLE `inventory`.`product_meta` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `productId` BIGINT NOT NULL,
  `key` VARCHAR(50) NOT NULL,
  `content` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  INDEX `idx_meta_product` (`productId` ASC),
  UNIQUE INDEX `uq_product_meta` (`productId` ASC, `key` ASC),
  CONSTRAINT `fk_meta_product`
    FOREIGN KEY (`productId`)
    REFERENCES `inventory`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Brand:

```
CREATE TABLE `inventory`.`brand` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `title` VARCHAR(75) NOT NULL,
  `summary` TINYTEXT NULL,
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NULL DEFAULT NULL,
  `content` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

Order:

```
CREATE TABLE `inventory`.`order` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `userId` BIGINT NOT NULL,
  `type` SMALLINT(6) NOT NULL DEFAULT 0,
  `status` SMALLINT(6) NOT NULL DEFAULT 0,
  `subTotal` FLOAT NOT NULL DEFAULT 0,
  `itemDiscount` FLOAT NOT NULL DEFAULT 0,
  `tax` FLOAT NOT NULL DEFAULT 0,
  `shipping` FLOAT NOT NULL DEFAULT 0,
```

```

`total` FLOAT NOT NULL DEFAULT 0,
`promo` VARCHAR(50) NULL DEFAULT NULL,
`discount` FLOAT NOT NULL DEFAULT 0,
`grandTotal` FLOAT NOT NULL DEFAULT 0,
`createdAt` DATETIME NOT NULL,
`updatedAt` DATETIME NULL DEFAULT NULL,
`content` TEXT NULL DEFAULT NULL,
PRIMARY KEY (`id`),
INDEX `idx_order_user` (`userId` ASC),
CONSTRAINT `fk_order_user`
  FOREIGN KEY (`userId`)
    REFERENCES `inventory`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

Order Item

```

CREATE TABLE `inventory`.`order_item` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `productId` BIGINT NOT NULL,
  `itemId` BIGINT NOT NULL,
  `orderId` BIGINT NOT NULL,
  `sku` VARCHAR(100) NOT NULL,
  `price` FLOAT NOT NULL DEFAULT 0,
  `discount` FLOAT NOT NULL DEFAULT 0,
  `quantity` SMALLINT(6) NOT NULL DEFAULT 0,
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NULL DEFAULT NULL,
  `content` TEXT NULL DEFAULT NULL,
PRIMARY KEY (`id`),
INDEX `idx_order_item_product` (`productId` ASC),
CONSTRAINT `fk_order_item_product`
  FOREIGN KEY (`productId`)
    REFERENCES `inventory`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

```

ALTER TABLE `inventory`.`order_item`
ADD INDEX `idx_order_item_item` (`itemId` ASC);
ALTER TABLE `inventory`.`order_item`
ADD CONSTRAINT `fk_order_item_item`
  FOREIGN KEY (`itemId`)
    REFERENCES `inventory`.`item` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;

```

```

ALTER TABLE `inventory`.`order_item`
ADD INDEX `idx_order_item_order` (`orderId` ASC);
ALTER TABLE `inventory`.`order_item`
ADD CONSTRAINT `fk_order_item_order`
  FOREIGN KEY (`orderId`)
    REFERENCES `inventory`.`order` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION;

```

Item:

```

CREATE TABLE `inventory`.`item` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,

```

```

`productId` BIGINT NOT NULL,
`brandId` BIGINT NOT NULL,
`supplierId` BIGINT NOT NULL,
`orderId` BIGINT NOT NULL,
`sku` VARCHAR(100) NOT NULL,
`mrp` FLOAT NOT NULL DEFAULT 0,
`discount` FLOAT NOT NULL DEFAULT 0,
`price` FLOAT NOT NULL DEFAULT 0,
`quantity` SMALLINT(6) NOT NULL DEFAULT 0,
`sold` SMALLINT(6) NOT NULL DEFAULT 0,
`available` SMALLINT(6) NOT NULL DEFAULT 0,
`defective` SMALLINT(6) NOT NULL DEFAULT 0,
`createdBy` BIGINT NOT NULL,
`updatedBy` BIGINT DEFAULT NULL,
`createdAt` DATETIME NOT NULL,
`updatedAt` DATETIME NULL DEFAULT NULL,
PRIMARY KEY (`id`),
INDEX `idx_item_product` (`productId` ASC),
CONSTRAINT `fk_item_product`
  FOREIGN KEY (`productId`)
  REFERENCES `inventory`.`product` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);

ALTER TABLE `inventory`.`item`
ADD INDEX `idx_item_brand` (`brandId` ASC);
ALTER TABLE `inventory`.`item`
ADD CONSTRAINT `fk_item_brand`
  FOREIGN KEY (`brandId`)
  REFERENCES `inventory`.`brand` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

ALTER TABLE `inventory`.`item`
ADD INDEX `idx_item_user` (`supplierId` ASC);
ALTER TABLE `inventory`.`item`
ADD CONSTRAINT `fk_item_user`
  FOREIGN KEY (`supplierId`)
  REFERENCES `inventory`.`user` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

ALTER TABLE `inventory`.`item`
ADD INDEX `idx_item_order` (`orderId` ASC);
ALTER TABLE `inventory`.`item`
ADD CONSTRAINT `fk_item_order`
  FOREIGN KEY (`orderId`)
  REFERENCES `inventory`.`order` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

```

Future Of Database:

The database is made using mysql. We can use the database to manage any inventory system like Food warehousing, Cosmetics storing, Raw products storing and many more. Followings are the goal that needs to be done in future:

1. Add transaction system.

2. Add specified suppliers' details.
3. Find out more efficient data on inventory.
4. Adding payment gateway.

Schema Diagram

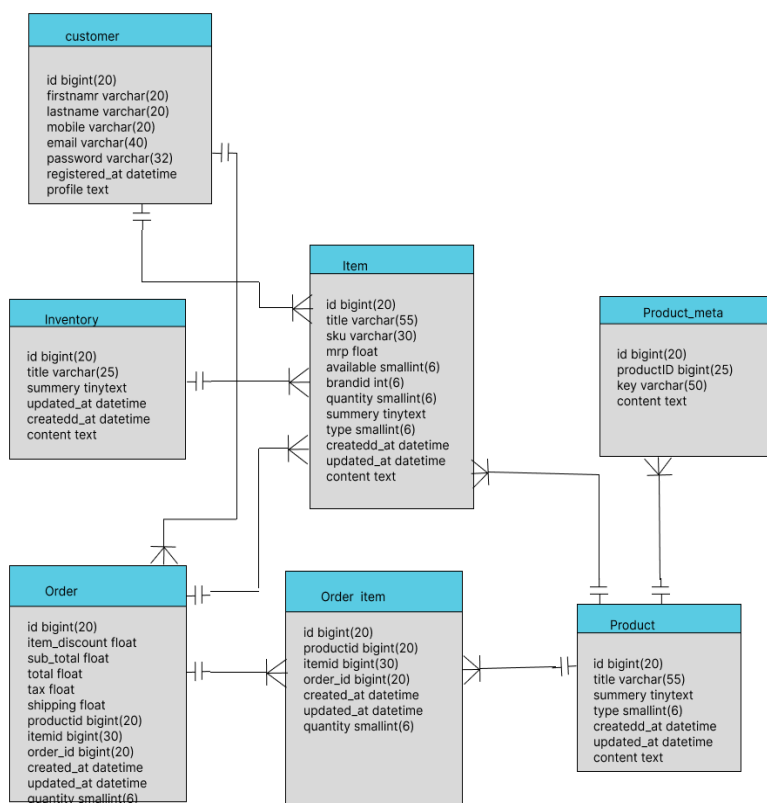


Fig: Schema Diagram Of Inventory Management System