INVENTORY MANAGEMENT SYSTEM (BEGINNER-LEVEL)

INVENTORY DATABASE:

The very first step is to create the Inventory Database. It can be created using the query as shown below.

```
CREATE SCHEMA `inventory` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

I have used the character set **utf8mb4** to support a wide range of characters.

User Table:

```
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 Table”

The Product Table with the appropriate constraints is shown below.

```
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;

The Product Meta Table with the appropriate constraints is as shown below.

```
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;
```

Category Table and Product Category Table:

The Category Table with the appropriate constraints is as shown below.

```
CREATE TABLE `inventory`.`category` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `parentId` BIGINT NULL DEFAULT NULL,
  `title` VARCHAR(75) NOT NULL,
  `metaTitle` VARCHAR(100) NULL DEFAULT NULL,
  `slug` VARCHAR(100) NOT NULL,
  `content` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`id`));

ALTER TABLE `inventory`.`category`
ADD INDEX `idx_category_parent` (`parentId` ASC);
ALTER TABLE `inventory`.`category`
ADD CONSTRAINT `fk_category_parent`
  FOREIGN KEY (`parentId`)
  REFERENCES `inventory`.`category` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;
```

The Product Category Table with the appropriate constraints is as shown below.

```
CREATE TABLE `inventory`.`product_category` (
  `productId` BIGINT NOT NULL,
  `categoryId` BIGINT NOT NULL,
  PRIMARY KEY (`productId`, `categoryId`),
  INDEX `idx_pc_category` (`categoryId` ASC),
  INDEX `idx_pc_product` (`productId` ASC),
  CONSTRAINT `fk_pc_product`
    FOREIGN KEY (`productId`)
    REFERENCES `inventory`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_pc_category`
    FOREIGN KEY (`categoryId`)
    REFERENCES `inventory`.`category` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

Brand Table:

The Brand Table with the appropriate constraints is shown below.

```
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 Table Table:

The Order Table with the appropriate constraints is as shown below.

```
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);
```

Address Table:

The Address Table with the appropriate constraints is as shown below.

```
CREATE TABLE `inventory`.`address` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `userId` BIGINT NULL DEFAULT NULL,
  `orderId` BIGINT NULL DEFAULT NULL,
  `firstName` VARCHAR(50) NULL DEFAULT NULL,
  `middleName` VARCHAR(50) NULL DEFAULT NULL,
  `lastName` VARCHAR(50) NULL DEFAULT NULL,
  `mobile` VARCHAR(15) NULL,
  `email` VARCHAR(50) NULL,
  `line1` VARCHAR(50) NULL DEFAULT NULL,
  `line2` VARCHAR(50) NULL DEFAULT NULL,
  `city` VARCHAR(50) NULL DEFAULT NULL,
  `province` VARCHAR(50) NULL DEFAULT NULL,
  `country` VARCHAR(50) NULL DEFAULT NULL,
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  INDEX `idx_address_user` (`userId` ASC),
  CONSTRAINT `fk_address_user`
    FOREIGN KEY (`userId`)
    REFERENCES `inventory`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

ALTER TABLE `inventory`.`address`
ADD INDEX `idx_address_order` (`orderId` ASC);
```

```
ALTER TABLE `inventory`.`address`
ADD CONSTRAINT `fk_address_order`
  FOREIGN KEY (`orderId`)
  REFERENCES `inventory`.`order` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;
```

Item Table:

The Item Table with the appropriate constraints is as shown below.

```
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;
```

Order Item Table:

The Order Item Table with the appropriate constraints is as shown below.

```
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;
```

Transaction Table:

```
CREATE TABLE `inventory`.`transaction` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `userId` BIGINT NOT NULL,
  `orderId` BIGINT NOT NULL,
  `code` VARCHAR(100) NOT NULL,
  `type` SMALLINT(6) NOT NULL DEFAULT 0,
  `mode` SMALLINT(6) NOT NULL DEFAULT 0,
  `status` 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_transaction_user` (`userId` ASC),
  CONSTRAINT `fk_transaction_user`
    FOREIGN KEY (`userId`)
    REFERENCES `inventory`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
ALTER TABLE `inventory`.`transaction`
ADD INDEX `idx_transaction_order` (`orderId` ASC);
ALTER TABLE `inventory`.`transaction`
ADD CONSTRAINT `fk_transaction_order`
  FOREIGN KEY (`orderId`)
  REFERENCES `inventory`.`order` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;
```