generator client {

provider = "prisma-client-js"

}

datasource db {

provider = "postgresql"

url = env("DATABASE\_URL")

}

model Asset {

assetId String @id @default(cuid()) @map("asset\_id")

type String

model String

serialNumber String @unique @map("serial\_number")

barcode String? @unique

status AssetStatus @default(AVAILABLE)

location String

purchaseDate DateTime @map("purchase\_date")

warrantyExpiry DateTime? @map("warranty\_expiry")

ownerHolderId String? @map("owner\_holder\_id")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

owner AssetHolder? @relation(fields: [ownerHolderId], references: [holderId], onDelete: SetNull)

assignments Assignment[]

shipments Shipment[]

maintenances Maintenance[]

returnRequests ReturnRequest[]

@@map("assets")

}

model AssetHolder {

holderId String @id @default(cuid()) @map("holder\_id")

name String

department String

email String @unique

location String

role HolderRole @default(EMPLOYEE)

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

ownedAssets Asset[]

assignments Assignment[]

shipments Shipment[]

returnRequests ReturnRequest[]

@@map("asset\_holders")

}

model Shipment {

shipmentId String @id @default(cuid()) @map("shipment\_id")

assetId String @map("asset\_id")

holderId String @map("holder\_id")

shipmentDate DateTime @map("shipment\_date")

deliveryStatus DeliveryStatus @default(IN\_TRANSIT) @map("delivery\_status")

carrier String

trackingNumber String? @map("tracking\_number")

courierApiResponse Json? @map("courier\_api\_response")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

asset Asset @relation(fields: [assetId], references: [assetId], onDelete: Cascade)

holder AssetHolder @relation(fields: [holderId], references: [holderId], onDelete: Cascade)

@@map("shipments")

}

model Assignment {

assignmentId String @id @default(cuid()) @map("assignment\_id")

assetId String @map("asset\_id")

holderId String @map("holder\_id")

assignedDate DateTime @map("assigned\_date")

returnDate DateTime? @map("return\_date")

status AssignmentStatus @default(ACTIVE)

acknowledgment Boolean @default(false)

chainOfCustodyDoc String? @map("chain\_of\_custody\_doc")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

asset Asset @relation(fields: [assetId], references: [assetId], onDelete: Cascade)

holder AssetHolder @relation(fields: [holderId], references: [holderId], onDelete: Cascade)

@@map("assignments")

}

model Maintenance {

maintenanceId String @id @default(cuid()) @map("maintenance\_id")

assetId String @map("asset\_id")

maintenanceDate DateTime @map("maintenance\_date")

issueReported String @map("issue\_reported")

resolution String?

warrantyCovered Boolean @default(false) @map("warranty\_covered")

reminderScheduled DateTime? @map("reminder\_scheduled")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

asset Asset @relation(fields: [assetId], references: [assetId], onDelete: Cascade)

@@map("maintenances")

}

model ReturnRequest {

returnId String @id @default(cuid()) @map("return\_id")

holderId String? @map("holder\_id")

customerId String? @map("customer\_id")

assetId String? @map("asset\_id")

orderId String? @map("order\_id")

requestDate DateTime @map("request\_date")

returnStatus ReturnStatus @default(PENDING) @map("return\_status")

prepaidLabelGenerated Boolean @default(false) @map("prepaid\_label\_generated")

finalStatusConfirmed Boolean @default(false) @map("final\_status\_confirmed")

reason String?

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

holder AssetHolder? @relation(fields: [holderId], references: [holderId], onDelete: SetNull)

customer Customer? @relation(fields: [customerId], references: [customerId], onDelete: SetNull)

asset Asset? @relation(fields: [assetId], references: [assetId], onDelete: SetNull)

order Order? @relation(fields: [orderId], references: [orderId], onDelete: SetNull)

@@map("return\_requests")

}

model AnalyticsReport {

reportId String @id @default(cuid()) @map("report\_id")

generatedDate DateTime @map("generated\_date")

reportType String @map("report\_type")

details Json

lostOrDamagedAssets Json? @map("lost\_or\_damaged\_assets")

complianceFlag Boolean @default(true) @map("compliance\_flag")

createdAt DateTime @default(now()) @map("created\_at")

@@map("analytics\_reports")

}

model ApiIntegration {

integrationId String @id @default(cuid()) @map("integration\_id")

systemName String @map("system\_name")

apiType String @map("api\_type")

endpointUrl String @map("endpoint\_url")

status IntegrationStatus @default(ACTIVE)

lastSyncTime DateTime? @map("last\_sync\_time")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

@@map("api\_integrations")

}

model Customer {

customerId String @id @default(cuid()) @map("customer\_id")

name String

email String @unique

phone String?

address String

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

orders Order[]

returnRequests ReturnRequest[]

@@map("customers")

}

model Product {

productId String @id @default(cuid()) @map("product\_id")

name String

description String?

category String

price Decimal @db.Decimal(10, 2)

status ProductStatus @default(AVAILABLE)

stockQuantity Int @default(0) @map("stock\_quantity")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

orderItems OrderItem[]

@@map("products")

}

model Order {

orderId String @id @default(cuid()) @map("order\_id")

customerId String @map("customer\_id")

orderDate DateTime @map("order\_date")

orderStatus OrderStatus @default(PENDING) @map("order\_status")

totalAmount Decimal @db.Decimal(10, 2) @map("total\_amount")

createdAt DateTime @default(now()) @map("created\_at")

updatedAt DateTime @updatedAt @map("updated\_at")

// Relations

customer Customer @relation(fields: [customerId], references: [customerId], onDelete: Cascade)

orderItems OrderItem[]

returnRequests ReturnRequest[]

@@map("orders")

}

model OrderItem {

orderItemId String @id @default(cuid()) @map("order\_item\_id")

orderId String @map("order\_id")

productId String @map("product\_id")

quantity Int

price Decimal @db.Decimal(10, 2)

createdAt DateTime @default(now()) @map("created\_at")

// Relations

order Order @relation(fields: [orderId], references: [orderId], onDelete: Cascade)

product Product @relation(fields: [productId], references: [productId], onDelete: Cascade)

@@map("order\_items")

}

// Enums

enum AssetStatus {

ASSIGNED

IN\_MAINTENANCE

AVAILABLE

RETIRED

LOST

}

enum HolderRole {

EMPLOYEE

MANAGER

ADMIN

DEPARTMENT

}

enum DeliveryStatus {

PENDING

IN\_TRANSIT

DELIVERED

FAILED

RETURNED

}

enum AssignmentStatus {

ACTIVE

RETURNED

EXPIRED

}

enum ReturnStatus {

PENDING

APPROVED

REJECTED

COMPLETED

CANCELLED

}

enum IntegrationStatus {

ACTIVE

INACTIVE

ERROR

}

enum ProductStatus {

AVAILABLE

OUT\_OF\_STOCK

DISCONTINUED

}

enum OrderStatus {

PENDING

CONFIRMED

SHIPPED

DELIVERED

CANCELLED

RETURNED

}