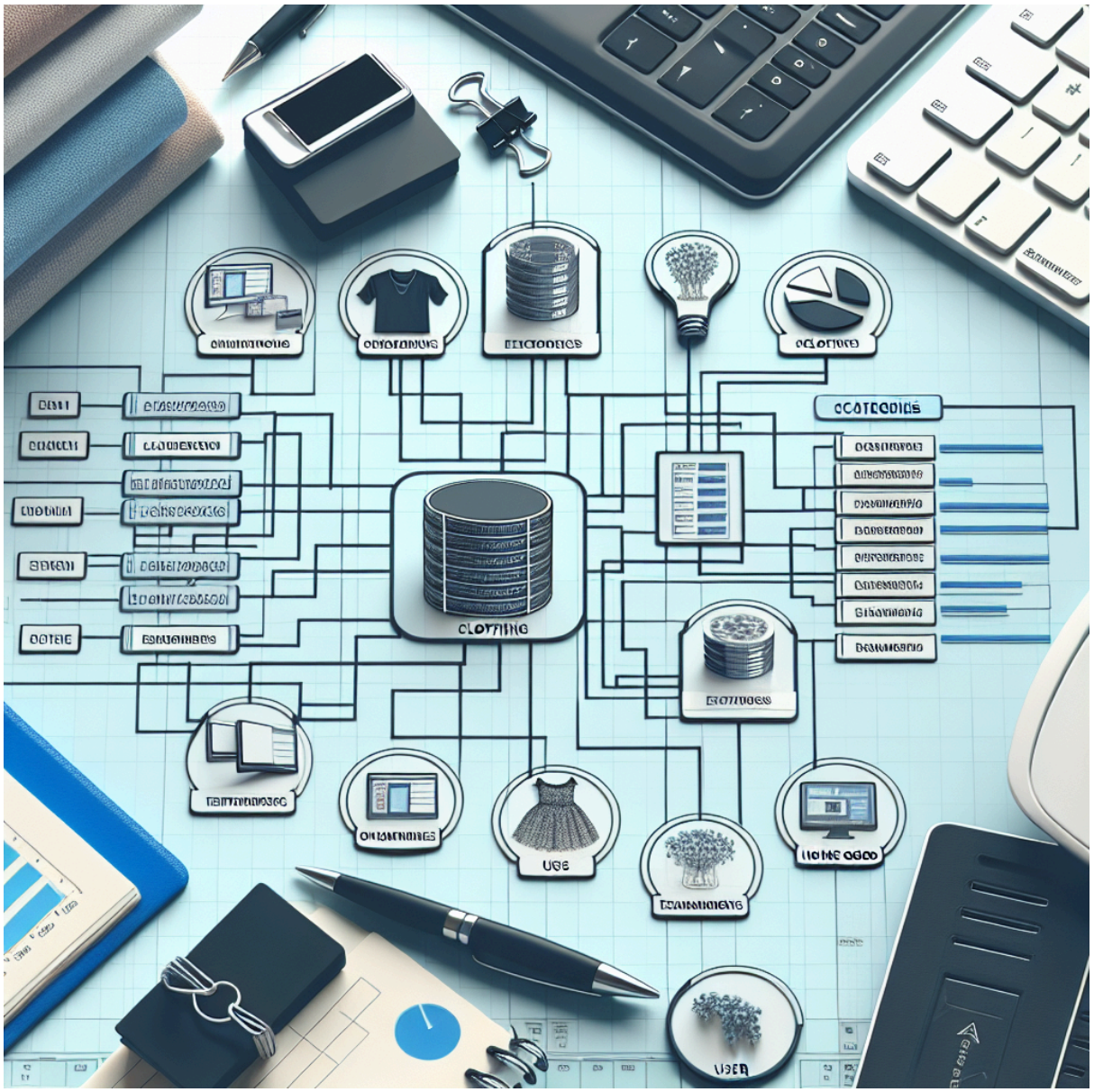


Databasmodellering
Eshop

2025-02-22

av

Seyed Mostafa Mohseni

TABLE OF CONTENT

Introduktion

En databasmodell av egen shop enligt kokboken [1].

Konceptuell modellering (kmom03)

1. Beskriv databasen i ett textstycke

Databasen är utformad för att hantera en e-handelsplattform där kunder köper produkter, och dessa transaktioner registreras. Både kund- och produktinformation lagras i systemet.

Den innehåller ett **kundregister** som lagrar kontaktuppgifter och ett **produktregister** där varje produkt har en unik produktkod, namn, beskrivning, pris och tillhör en eller flera **produktkategorier**.

Ett **lagerhanteringssystem** håller reda på produkternas tillgänglighet(antal) samt deras placering i lagret(hylla), där samma produkt kan vara fördelad på flera hyllor.

När en kund gör en beställning skapas en **order**, som lagrar information om kunden, de beställda produkterna och dess beställda antal.

Utifrån ordern skickas en **plocklista** till lagret där produkterna matchas med deras lagerhyllor för att underlätta orderhanteringen.

Efter att ordern har packats, bifogas en **faktura**, som inkluderar en summering av köpet, inklusive pris per produkt och totalbelopp.

Slutligen innehåller databasen en **logg** där viktiga händelser, såsom skapande och borttagning av ordrar och fakturor, registreras.

Databasen är utformad för att kunna integreras med både ett **webbinterface** för kunder och ett **terminalinterface** för administrativ hantering.

2. Skriv ned alla entiteter

- **Kundregister*(customer)***:KundID,Förnamn,Efternamn,E-post,Adress,Telefonnummer.
- **Produktregister*(product)***: ProduktID, Produktnamn, Beskrivning, Pris,Lagerkvantitet.
- **Produktkategorier*(category)***: KategoriID, Kategorinamn.
- **Lager*(stock)***: LagerID, ProduktID, Lagerkvantitet, Hyllplacering.
- **Orderhantering*(order)***: OrderID, KundID, Orderdatum, OrderStatus, Totalpris.
- **Produkter i en order*(orderItem)***: OrderradID, OrderID, ProduktID, Antal, Pris.
- **Plocklista*(delivery)***: Leverans-ID, OrderID, Leveransdatum, Status.
- **Fakturering*(invoice)***: FakturaID, OrderID, Faktureringsdatum,Totalpris.
- **Logg*(inventoryEvents)***: EventID, Eventbeskrivning, Eventdatum.
- **Kopplingstabell för Produkt och Kategori(ProductCategory):** ProduktID,KategoriID

3. Skriv ned alla relationer och visa i matris

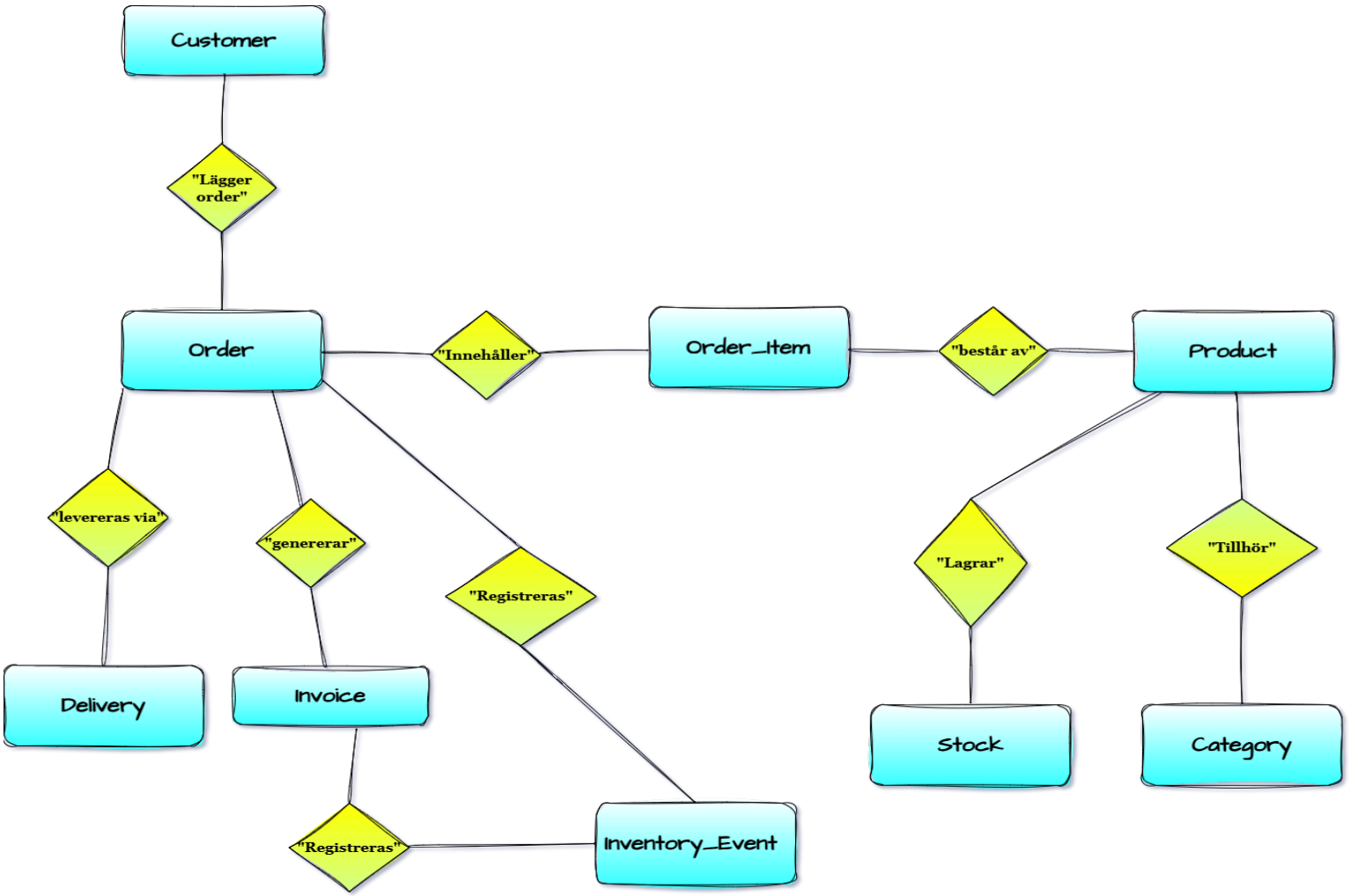
- **Customer → Order:** En kund kan ha en eller flera ordrar.(1:N)
- **Order → Customer:** En order är kopplad till en specifik kund.(N:1)
- **Product → Product_Category:** En produkt kan tillhöra flera kategorier.(1:N)
- **Category → Product_Category:** En kategori kan ha en eller flera produkter.(1:N)
- **Product → Stock:** En produkt kan finnas på en eller flera lagerhyllor.(1:N)
- **Stock → Product:** En lagerplats är kopplad till en specifik produkt.(N:1)
- **Product → Order_Item:** En produkt kan förekomma i en eller flera orderrader.(1:N)
- **Order_Item → Product:** En orderrad är kopplad till en specifik produkt.(N:1)
- **Order → Order_Item:** En order innehåller en eller flera orderrader.(1:N)
- **Order_Item → Order:** En orderrad tillhör en specifik order.(N:1)
- **Order → Delivery:** En order kan ha en eller flera plocklistor kopplade till sig.(1:N)
- **Delivery → Order:** En plocklista är kopplad till en specifik order.(N:1)
- **Order → Invoice:** En order är kopplad till en faktura.(1:1)
- **Invoice → Order:** En faktura är kopplad till en specifik order.(1:1)
- **Inventory_Event → Order:** En eller flera händelser kan vara kopplade till en order.(N:1)
- **Inventory_Event → Invoice:** En eller flera händelser kan vara kopplade till en faktura.(N:1)

Denna beskrivning sammanfattar hur entiteterna i databasen är relaterade till varandra!

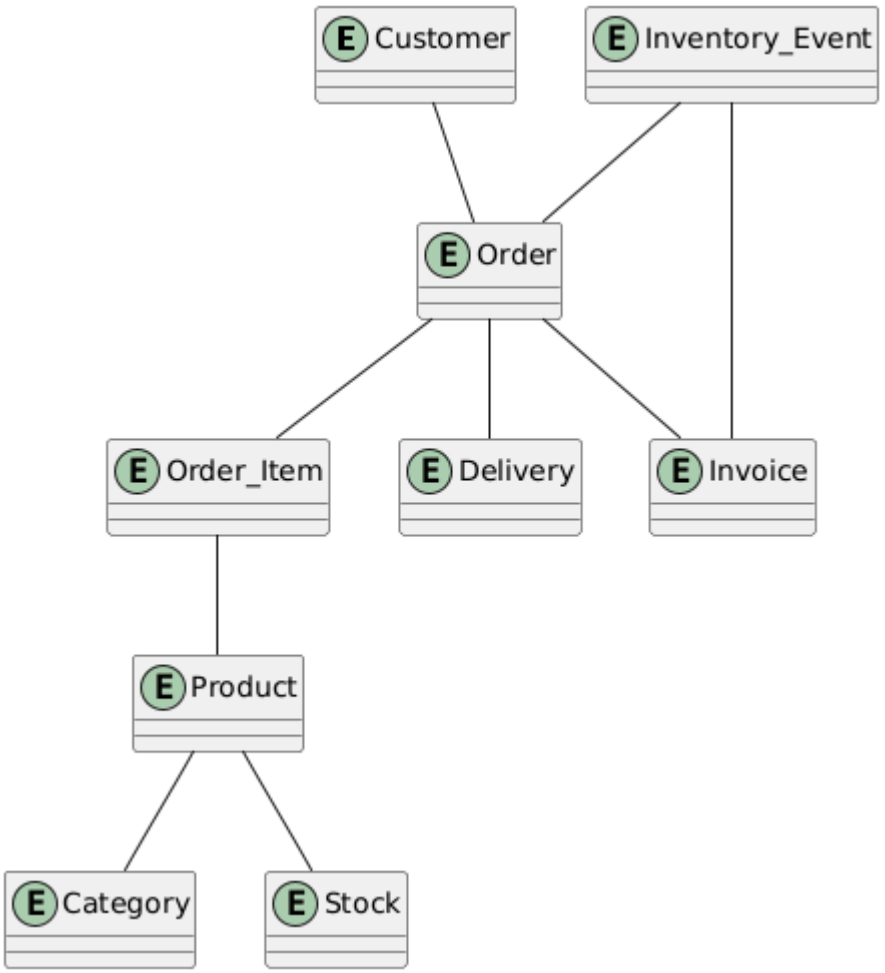
Matris:

Relationer	**Customer**	**Product**	**Category**	**Stock**	**Order**	**Order_ Item**	**Delivery**	**Invoice**	**Inventory_ Event**
Customer					har en/flera				
Product			finns i en/flera kategorier	finns på en/flera hyllor		finns på en/flera orderrader			
Category		har en/flera produkter							
Stock		en lagerplats är kopplad till en produkt							
Order	en order är kopplad till en kund					har en/flera rader	har en/flera plocklista	en order är kopplad till en faktura	
Order_Item		en rad har en produkt			en rad tillhör en order				
Delivery					en plocklista kopplad till en order				
Invoice					en faktura kopplad till en order				
Inventory_ Event					en/flera event kopplad till en order			en/flera event kopplad till en faktura	

4. Rita enkelt ER-diagram med entiteter och relationer

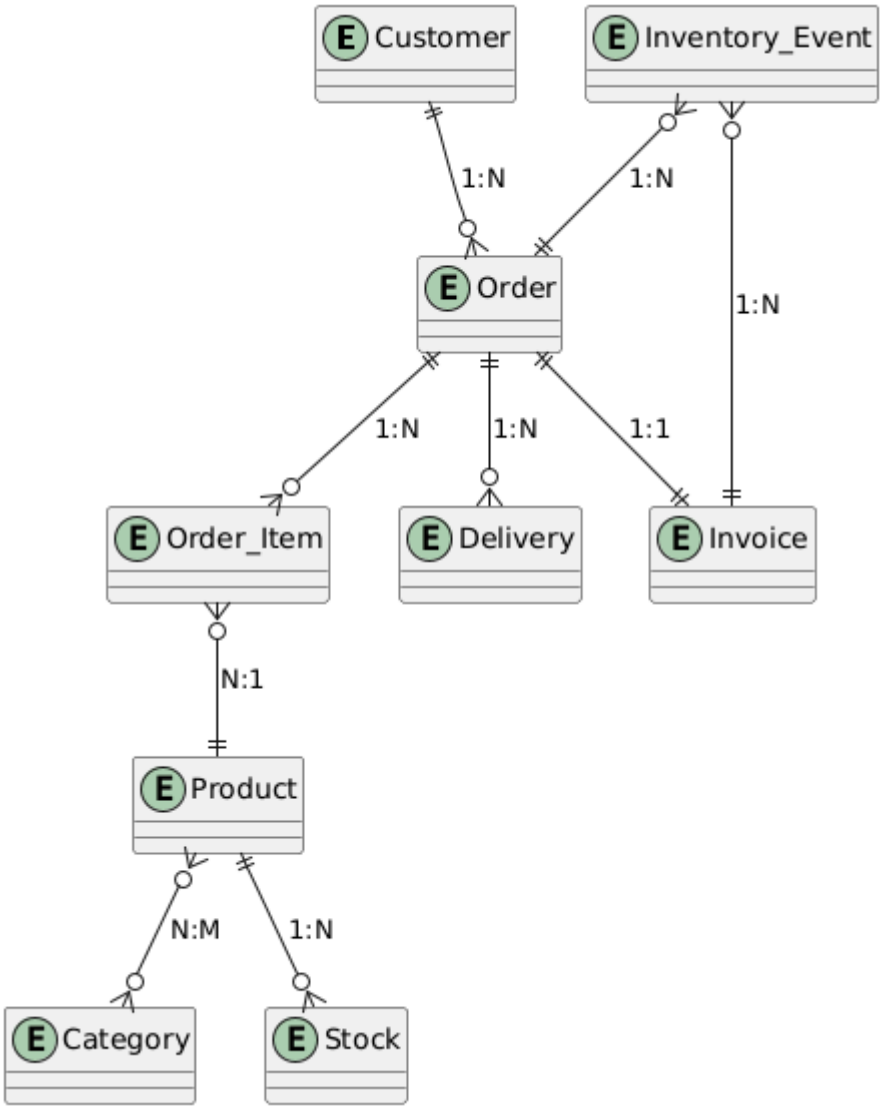


I plantuml:


Customer
Inventory_Event
Order
Order_Item
Delivery
Invoice
Product
Category
Stock

5. Komplettera ER-diagram med kardinalitet

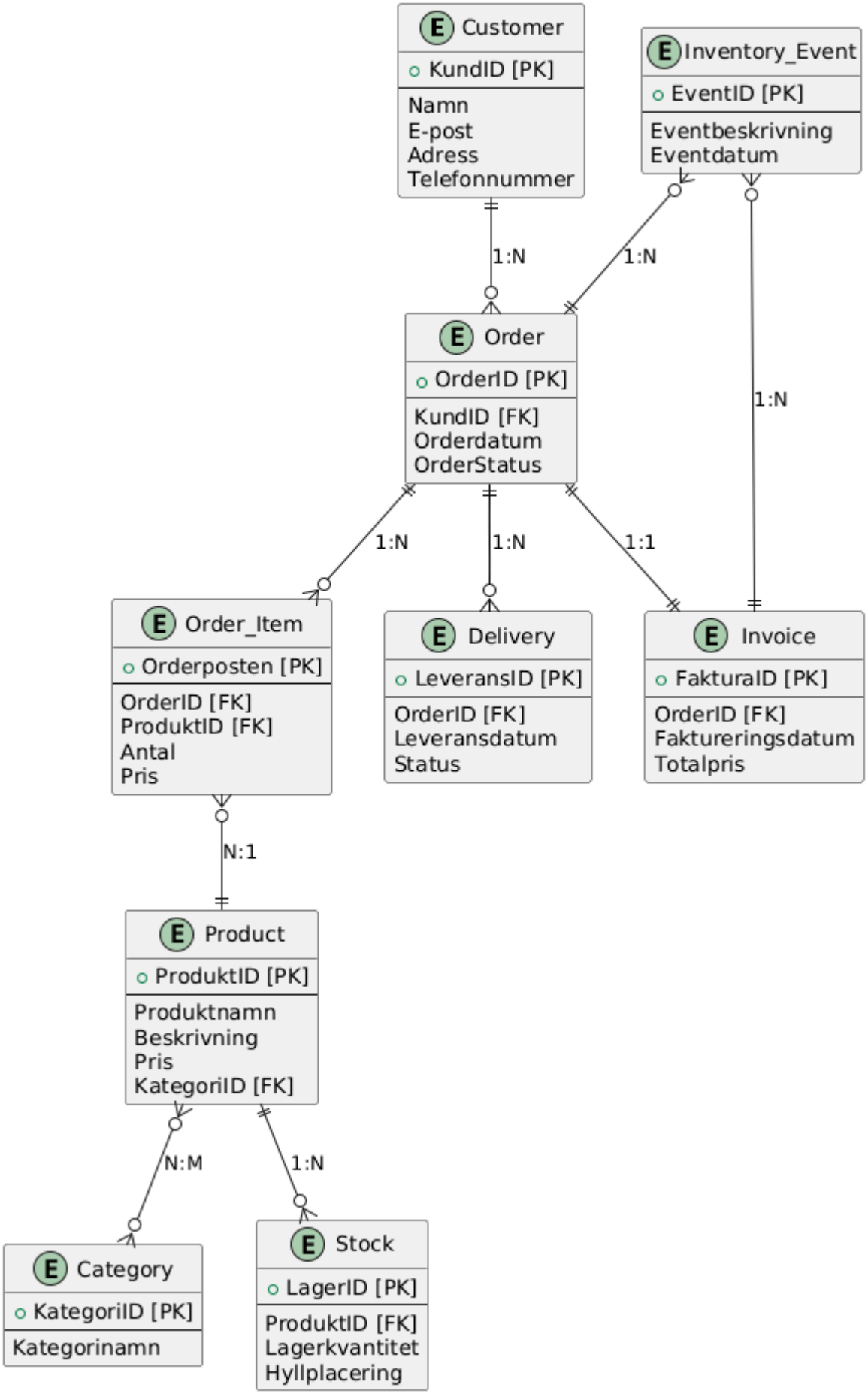
**Förklaring*(rak linje (||) indikerar “en”, och en cirkel med en gren ({) indikerar “många”)*



Beskrivning av relationerna(detta finns även i steg 3):
En kund kan ha flera ordrar (1:N).
En order kan innehålla flera orderrader (1:N).
En order har exakt en faktura (1:1).
En order kan ha flera plocklistor (1:N).
En orderrad är kopplad till en specifik produkt (N:1).
En produkt kan tillhöra flera kategorier, och en kategori kan innehålla flera produkter (1:N).
En produkt kan lagras på flera lagerplatser (1:N).
En order kan vara kopplad till flera händelser(1:N).
En faktura kan vara kopplade till flera händelser (1:N).

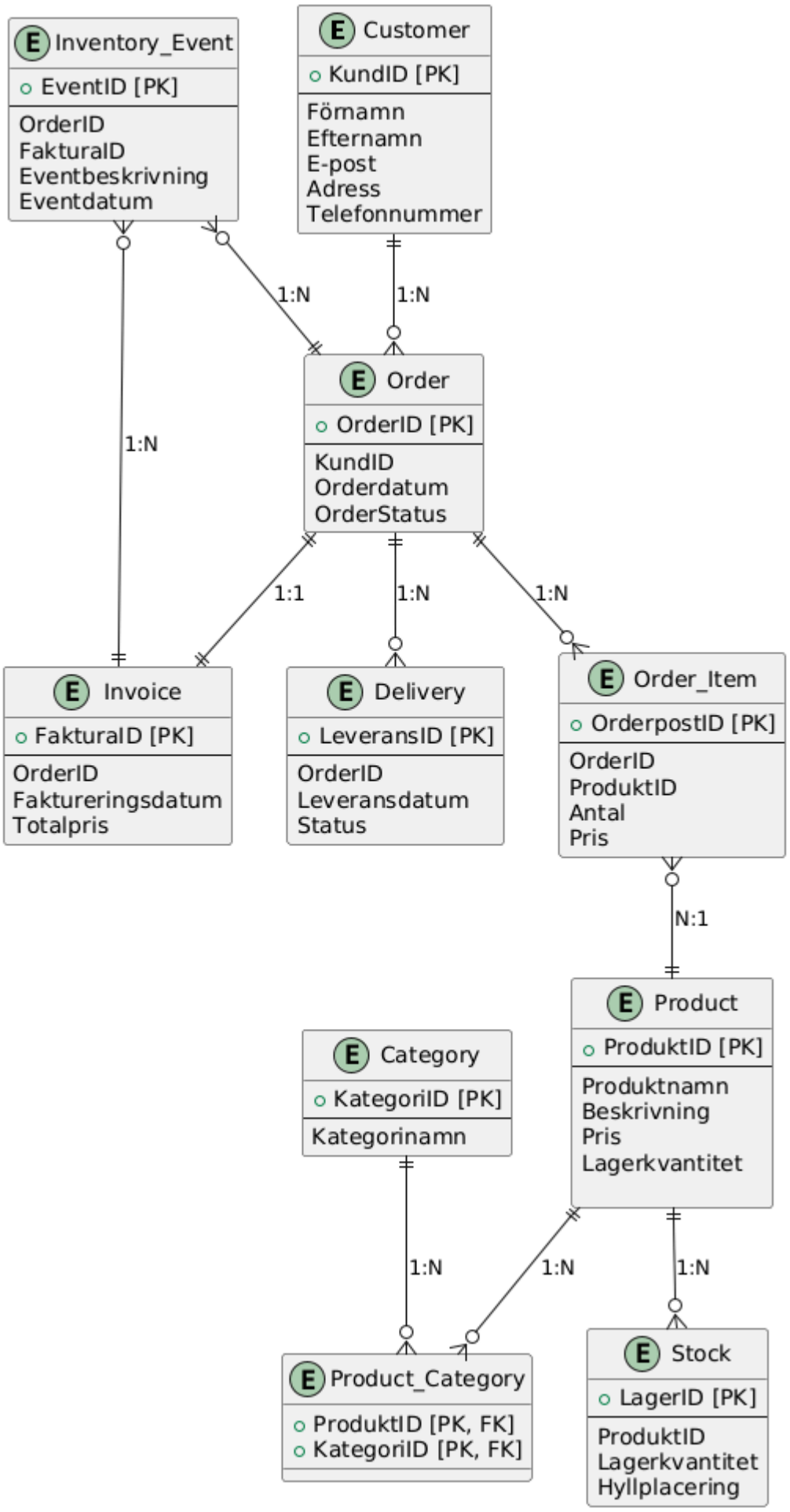
6. Komplettera ER-diagram med alla attribut samt kandidatnycklar

**Förklaring*(primärnycklar (PK) och främmande nycklar (FK))*


Customer
KundID [PK]
Namn
E-post
Adress
Telefonnummer
Inventory_Event
EventID [PK]
Eventbeskrivning
Eventdatum
Order
OrderID [PK]
KundID [FK]
Orderdatum
OrderStatus
Order_Item
Orderposten [PK]
OrderID [FK]
ProduktID [FK]
Antal
Pris
Delivery
LeveransID [PK]
OrderID [FK]
Leveransdatum
Status
Invoice
FakturaID [PK]
OrderID [FK]
Faktureringsdatum
Totalpris
Product
ProduktID [PK]
Produktnamn
Beskrivning
Pris
KategoriID [FK]
Category
KategoriID [PK]
Kategorinamn
Stock
LagerID [PK]
ProduktID [FK]
Lagerkvantitet
Hyllplacering
1:N
1:N
1:N
1:N
1:N
1:1
N:1
N:M
1:N

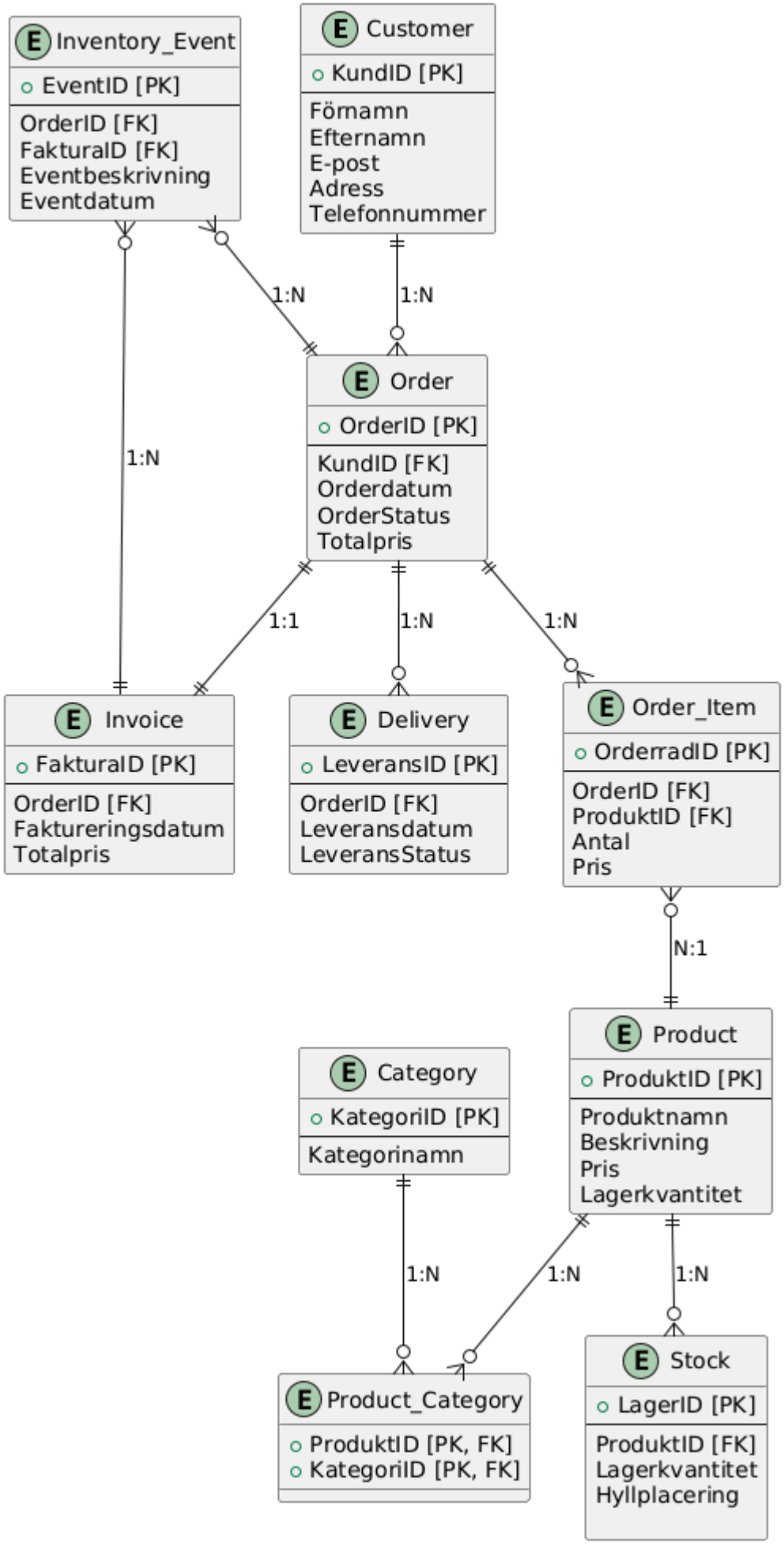
Logisk modellering (kmom04)

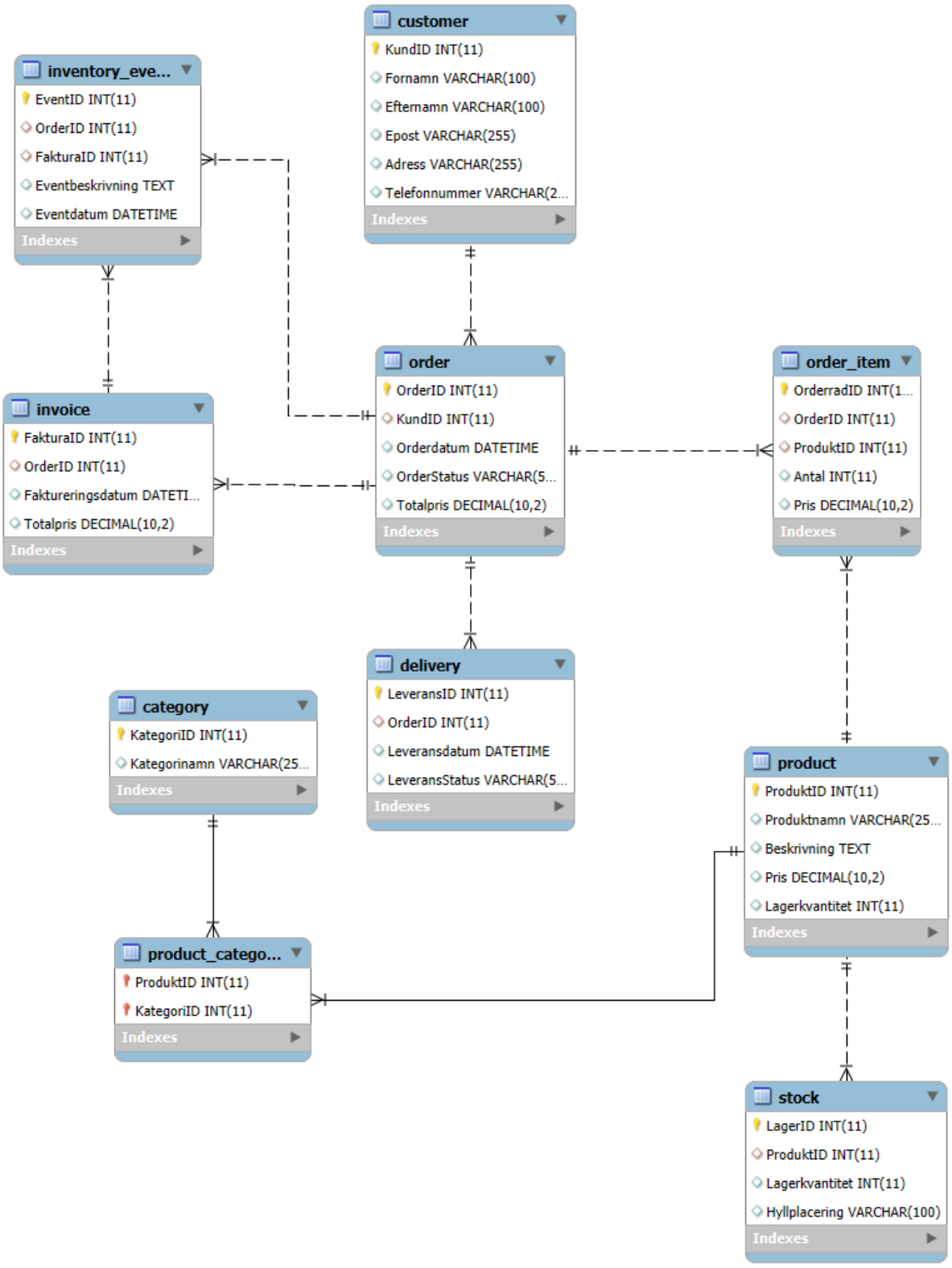
7. Modifiera ER-diagram enligt relationsmodellen


Inventory_Event
EventID [PK]
OrderID
FakturaID
Eventbeskrivning
Eventdatum
Customer
KundID [PK]
Förnamn
Efternamn
E-post
Adress
Telefonnummer
1:N
1:N
1:N
Order
OrderID [PK]
KundID
Orderdatum
OrderStatus
1:1
1:N
1:N
Invoice
FakturaID [PK]
OrderID
Faktureringsdatum
Totalpris
Delivery
LeveransID [PK]
OrderID
Leveransdatum
Status
Order_Item
OrderpostID [PK]
OrderID
ProduktID
Antal
Pris
N:1
Product
ProduktID [PK]
Produktnamn
Beskrivning
Pris
Lagerkvantitet
Category
KategoriID [PK]
Kategorinamn
1:N
1:N
1:N
Product_Category
ProduktID [PK, FK]
KategoriID [PK, FK]
Stock
LagerID [PK]
ProduktID
Lagerkvantitet
Hyllplacering

8. Utöka ER-diagram med primära/främmande nycklar samt kompletterande attribut

**Förklaring*(primärnycklar (PK) och främmande nycklar (FK))*


Inventory_Event
EventID [PK]
OrderID [FK]
FakturaID [FK]
Eventbeskrivning
Eventdatum
Customer
KundID [PK]
Förnamn
Efternamn
E-post
Adress
Telefonnummer
1:N
1:N
1:N
Order
OrderID [PK]
KundID [FK]
Orderdatum
OrderStatus
Totalpris
1:1
1:N
1:N
Invoice
FakturaID [PK]
OrderID [FK]
Faktureringsdatum
Totalpris
Delivery
LeveransID [PK]
OrderID [FK]
Leveransdatum
LeveransStatus
Order_Item
OrderradID [PK]
OrderID [FK]
ProduktID [FK]
Antal
Pris
N:1
Product
ProduktID [PK]
Produktnamn
Beskrivning
Pris
Lagerkvantitet
Category
KategoriID [PK]
Kategorinamn
1:N
1:N
1:N
Product_Category
ProduktID [PK, FK]
KategoriID [PK, FK]
Stock
LagerID [PK]
ProduktID [FK]
Lagerkvantitet
Hyllplacering


customer
KundID INT(11)
Fornamn VARCHAR(100)
Efternamn VARCHAR(100)
Epost VARCHAR(255)
Adress VARCHAR(255)
Telefonnummer VARCHAR(2...
Indexes
inventory_eve...
EventID INT(11)
OrderID INT(11)
FakturaID INT(11)
Eventbeskrivning TEXT
Eventdatum DATETIME
Indexes
order
OrderID INT(11)
KundID INT(11)
Orderdatum DATETIME
OrderStatus VARCHAR(5...
Totalpris DECIMAL(10,2)
Indexes
order_item
OrderradID INT(1...
OrderID INT(11)
ProduktID INT(11)
Antal INT(11)
Pris DECIMAL(10,2)
Indexes
invoice
FakturaID INT(11)
OrderID INT(11)
Faktureringsdatum DATETI...
Totalpris DECIMAL(10,2)
Indexes
delivery
LeveransID INT(11)
OrderID INT(11)
Leveransdatum DATETIME
LeveransStatus VARCHAR(5...
Indexes
category
KategoriID INT(11)
Kategorinamn VARCHAR(25...
Indexes
product
ProduktID INT(11)
Produktnamn VARCHAR(25...
Beskrivning TEXT
Pris DECIMAL(10,2)
Lagerkvantitet INT(11)
Indexes
product_catego...
ProduktID INT(11)
KategoriID INT(11)
Indexes
stock
LagerID INT(11)
ProduktID INT(11)
Lagerkvantitet INT(11)
Hyllplacering VARCHAR(100)
Indexes

Fysisk modellering (kmom04)

9. Skapa SQL DDL för tabellerna

Jag har skrivit all SQL DDL kod själv och noggrant skapat och strukturerat mina databastabeller. Alla 10 tabeller är nu färdiga med de nödvändiga primärnycklarna och främmande nycklarna för att säkerställa korrekta relationer mellan dem. För att göra koden lättillgänglig och organiserad har jag lagrat den i ddl.sql-filen. Dessutom har jag genomfört en reverse engineering-process för att visualisera och analysera tabellernas struktur och hur de är kopplade till varandra. Detta har gett mig en bättre förståelse av databasen och dess uppbyggnad.

10. Lista funktioner som databasen skall stödja (API)

Här är en lista över funktioner som databasen bör stödja genom dess API:

1. Registrering och hantering av kunder:
- Skapa en ny kundprofil. Radera kunder.
- Uppdatera befintliga kunduppgifter (t.ex. kontaktinformation).
- Hämta kundinformation baserat på kund-ID eller e-postadress.

2. Hantering av produkter:
- Lägga till nya produkter i produktregistret. Radera produkter.
- Uppdatera produktinformation (t.ex. namn, beskrivning, pris).
- Hämta produktinformation baserat på produkt-ID eller produktkategori.

3. Hantering av kategorier:
- Skapa nya kategorier och produktkategorier.
-Koppla produkter till kategorier.
- Uppdatera kategorinamn.
- Hämta kategorilista och dess tillhörande produkter.
- Ta bort kategorier.

4. Hantering av lager:
- Registrera händelser i lager (t.ex. mottagande av varor, flyttning av varor, justeringar av lagerstatus).
- Uppdatera lagerstatus för enskilda produkter.
- Hämta lagerstatus för produkter baserat på produkt-ID eller hyllplacering.

5. Orderhantering:
- Skapa en ny order för en kund. Avbryta order.
- Lägga till produkter i en befintlig order.
- Uppdatera orderstatus (t.ex. behandling, leverans, slutförd, avbruten).
- Hämta orderhistorik för en specifik kund eller order baserat på order-ID.
-lägga till/ta bort/uppdatera/visa rader i en order

6. Hantering av leveranser och fakturor:
- Skapa en ny leverans(plocklisat) från en befintlig order.
- Uppdatera leveransstatus (t.ex. packad, skickad, levererad).
- Skapa en faktura baserat på en befintlig order.
- Hämta leverans- och faktureringsinformation baserat på order-ID.

7. Loggning av händelser:
-Exportera orderhistorik och faktureringsinformation.
- Registrera händelser relaterade till kundinteraktioner (t.ex. inloggning, registrering, ändringar av uppgifter).
- Registrera händelser relaterade till orderhantering (t.ex. skapande, uppdatering, avbokning av order).
-Registrera händelser relaterade till fakturering(ny faktura eller borttagning av faktura)
- Hämta loggposter baserade på händelsetyp, användar-ID eller datumintervall.

Genom att implementera dessa funktioner kan API:et möjliggöra en komplett hantering av kunder, produkter, lager, order, leveranser och fakturor inom e-handelsplattformen.

APPENDIX DDL

```
--

-- Skapar databas

--



drop database if exists eshop;

CREATE DATABASE IF NOT EXISTS eshop;



-- Välj vilken databas du vill använda

USE eshop;
```

```
--

-- Table structure for database `eshop`

--



-- drop tables

DROP TABLE IF EXISTS Inventory_Event;

DROP TABLE IF EXISTS Invoice;
```

```sql
DROP TABLE IF EXISTS Delivery;

DROP TABLE IF EXISTS Order_Item;

DROP TABLE IF EXISTS Order;

DROP TABLE IF EXISTS Stock;

DROP TABLE IF EXISTS Product_Category;

DROP TABLE IF EXISTS Category;

DROP TABLE IF EXISTS Product;

DROP TABLE IF EXISTS Customer;




-- Kundregister

CREATE TABLE Customer (

    KundID INT PRIMARY KEY,

    Fornamn VARCHAR(100),

    Efternamn VARCHAR(100),

    Epost VARCHAR(255),

    Adress VARCHAR(255),

    Telefonnummer VARCHAR(20)

);
```

```sql
-- Produktregister
CREATE TABLE Product (
    ProduktID INT PRIMARY KEY,
    Produktnamn VARCHAR(255),
    Beskrivning TEXT,
    Pris DECIMAL(10,2),
    Lagerkvantitet INT
);

-- Kategori
CREATE TABLE Category (
    KategoriID INT PRIMARY KEY,
    Kategorinamn VARCHAR(255)
);

-- Kopplingstabell mellan produkt och kategori
CREATE TABLE Product_Category (
```

```
    ProduktID INT,

    KategoriID INT,

    PRIMARY KEY (ProduktID, KategoriID),

    FOREIGN KEY (ProduktID) REFERENCES Product(ProduktID),

    FOREIGN KEY (KategoriID) REFERENCES Category(KategoriID)

);




--lagerregister

CREATE TABLE Stock (

    LagerID INT PRIMARY KEY,

    ProduktID INT,

    Lagerkvantitet INT,

    Hyllplacering VARCHAR(100),

    FOREIGN KEY (ProduktID) REFERENCES Product(ProduktID)

);
```

```sql
-- Orderhantering

CREATE TABLE Order (

    OrderID INT PRIMARY KEY,

    KundID INT,

    Orderdatum DATETIME,

    OrderStatus VARCHAR(50), -- väntar, bearbetas, slutförd, avbruten

    Totalpris DECIMAL(10,2),

    FOREIGN KEY (KundID) REFERENCES Customer(KundID)

);




-- Produkter i en order

CREATE TABLE Order_Item (

    OrderradID INT PRIMARY KEY,

    OrderID INT,

    ProduktID INT,

    Antal INT,

    Pris DECIMAL(10,2),

    FOREIGN KEY (OrderID) REFERENCES Order(OrderID),
```

```
    FOREIGN KEY (ProduktID) REFERENCES Product(ProduktID)

);




-- leveranshantering och plocklista

CREATE TABLE Delivery (

    LeveransID INT PRIMARY KEY,

    OrderID INT,

    Leveransdatum DATETIME,

    LeveransStatus VARCHAR(50), -- packad, skickad, levererad

    FOREIGN KEY (OrderID) REFERENCES Order(OrderID)

);




-- fakturering

CREATE TABLE Invoice (

    FakturaID INT PRIMARY KEY,

    OrderID INT,

    Faktureringsdatum DATETIME,
```

```
    Totalpris DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES Order(OrderID)
);



-- lagerhändelser
CREATE TABLE Inventory_Event (
    EventID INT PRIMARY KEY,
    OrderID INT,
    FakturaID INT,
    Eventbeskrivning TEXT,
    Eventdatum DATETIME,
    FOREIGN KEY (OrderID) REFERENCES Order(OrderID),
    FOREIGN KEY (FakturaID) REFERENCES Invoice(FakturaID)
);
```

REFERENSER

[1] Kokbok för databasmodellering, https://dbwebb.se/kunskap/kokbok-for-databasmodellering, visited 2025-03-08.